Massachusetts Institute of Technology

Lincoln Laboratory *Journal*

Volume 23 ■ Number 1 ■ 2019

SERIOUS GAMES

LINCOLN LABORATORY *JOURNAL*
Volume 23 ■ Number 1 ■ 2019

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY *Journal*

The *Lincoln Laboratory Journal* (ISSN 0896-4130) is published twice a year by Massachusetts Institute of Technology, Lincoln Laboratory, 244 Wood Street, Lexington, MA 02421-6426. Subscriptions are free of charge, but provided only to qualified recipients (government employees and contractors, libraries, university faculty, and R&D laboratories). Requests for individual copies, subscriptions, or permission to reprint articles should be submitted to the Subscription Coordinator, Room S3-106, at the address above. The editorial offices can also be reached at 781-981-4204.

Journal Online: Selected articles from back issues are available at www.ll.mit.edu/about/lincoln-laboratory-publications/lincoln-laboratory-journal. Contents of the current issue will be posted on this site approximately 45 days following print publication.

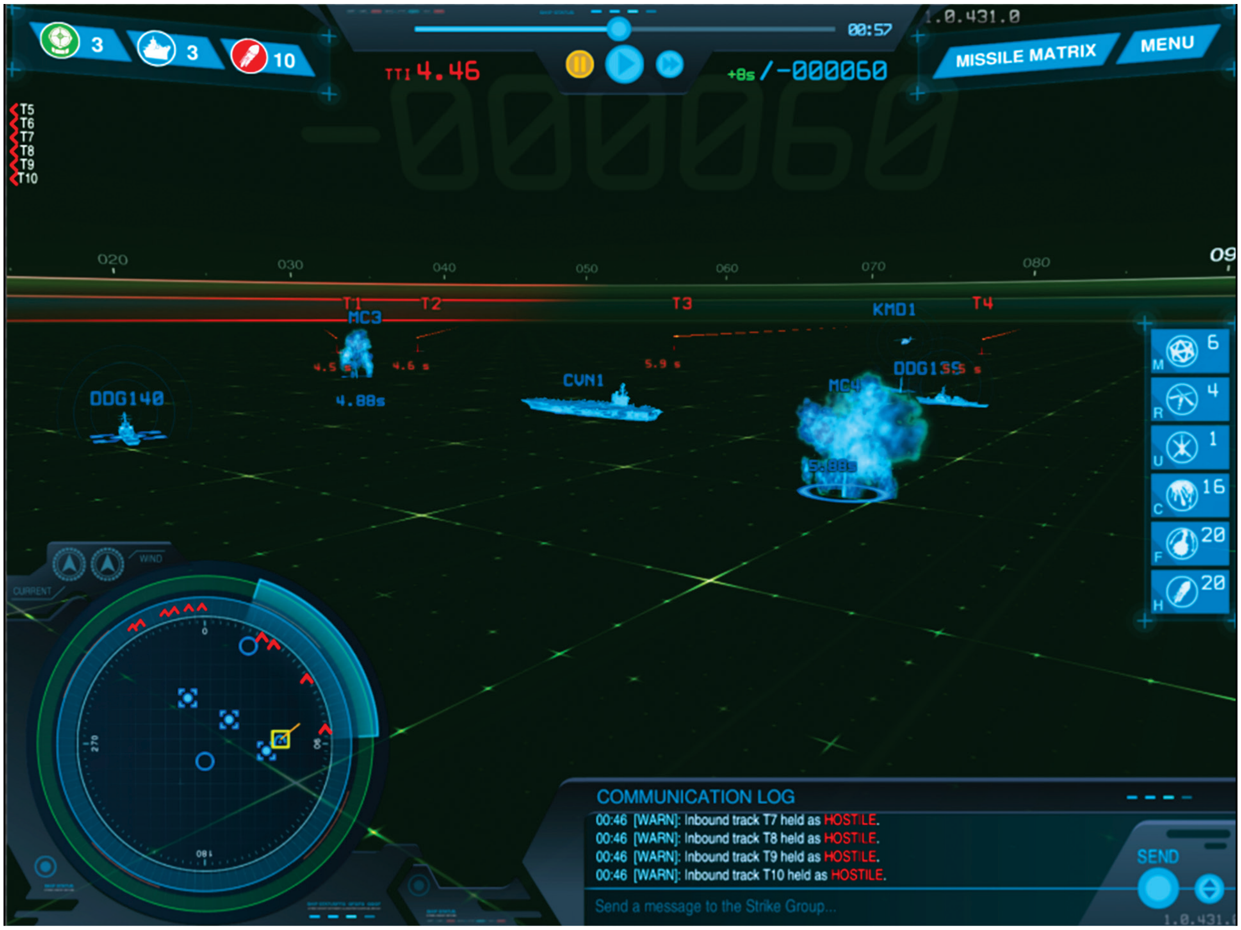
This material is based upon work supported by the U.S. Department of the Air Force under Air Force Contract no. FA8702-15-D-0001. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Air Force.

www.ll.mit.edu

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY *Journal*
Volume 23 ▪ Number 1 ▪ 2019
SERIOUS GAMES

On the Cover

Uncovering and thwarting a clandestine network that may be plotting terrorist acts is the mission attempted by players of the dark net discovery game. One element of this game is locating and tracking vehicles that may be involved in the terrorist plot. To achieve that goal, players interpret game-provided information to perform vehicle triage, homing in on vehicles traveling to locations presumed to be connected to suspicious activity. The cover image shows all vehicle track data in the game, with the red lines denoting tracks associated with suspected terrorist network vehicles and the yellow lines denoting an innocuous background of tracks for vehicles driven by the general population.



00:46 [WARN]: Inbound track T7 held as HOSTILE.
00:46 [WARN]: Inbound track T8 held as HOSTILE.
00:46 [WARN]: Inbound track T9 held as HOSTILE.
00:46 [WARN]: Inbound track T10 held as HOSTILE.

Send a message to the Strike Group...

The display for *Strike Group Defender* gameplay presents blue ships and red threats (center), an overhead view (lower left), a message panel (lower right), a countermeasure inventory (right), and menus and scoreboard (top). For more information, see the article “Strike Group Defender” on page 25.

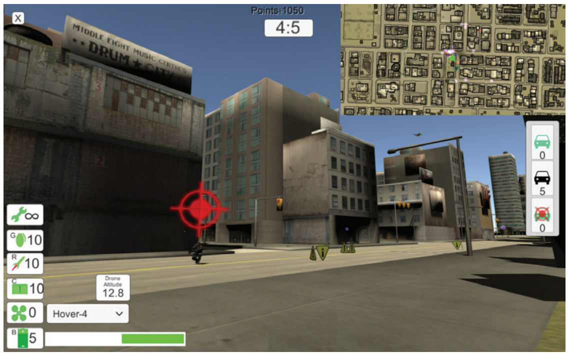
Massachusetts Institute of Technology

LINCOLN LABORATORY *Journal*

■ VOLUME 23, NUMBER 1, 2019 ■

Lab Notes

Features



Looking Back

Game-Based Human-System Analysis for National Security R&D

Matthew P. Daggett, Timothy J. Dasey, Adam S. Norige, and Robert M. Seater

Serious games are influencing efforts to improve education, health care, defense, and awareness of societal issues by applying gamification to help users develop understanding and skills in these fields and to elicit knowledge from expert users. Researchers at Lincoln Laboratory are transforming traditional research and development processes by using games to design, engineer, and assess more efficient and effective sociotechnical systems for national security needs.

Since the 1950s, MIT Lincoln Laboratory has conducted rigorous systems analysis, full-system prototyping, and development of long-term advanced technologies for national security applications. As the discrete systems of earlier decades have been replaced with complex interconnected systems of systems, traditional modeling and simulation and systems analysis often insufficiently account for human dynamics. These limitations become further exacerbated as a long-standing paradigm of systems as subordinate to operators is being replaced with collaborative workflows enabled by automation and artificial intelligence. To address these challenges, researchers at the Laboratory have developed methodologies and technologies for designing, building, and employing serious games that measure human decision making and that serve as systems analysis tools to assess and facilitate complex human-system dynamics that approximate those of realistic sociotechnical systems. These serious games are a unique tool in the research and development (R&D) process that overcomes the limitations of other methods.

Games are a structured form of play, usually undertaken for enjoyment, achievement, or reward. They have been recorded as part of cultures dating back to the 26th century BCE and are thought to be universal to the human experience. Games have also long been used for U.S. national security purposes, with some of the earliest wargame efforts in the 1800s at the Naval War College. Many early wargames were large tabletop or seminar-style formats used for developing war plans and exercising decision making. With the advent of modern computing, these games have grown into large, sophisticated, distributed semiautomated force simulations, largely focused on informing military training and doctrine.

In the 1970s, a definition of the term serious games emerged to broadly define games as a means to achieve an explicit purpose other than amusement. Under this rubric, gamification has been employed in education, scientific exploration, health care, emergency management, and more. While the use of games in a national security context is often synonymous with wargaming, Lincoln Laboratory's research into games aligns with the broader view of the application of serious games.

Gaming at Lincoln Laboratory

Since 2001, the Laboratory has been developing purpose-built serious games and applying them to its core R&D processes across a variety of mission areas, including air and missile defense; intelligence, surveillance, and reconnaissance; chemical and biological defense; air traffic management; cyber security operations; and emergency response. While built with different objectives, these games fall mainly into three common game applications:

1. Experiential learning. Games are a natural fit for training and can model situations that are rare in practice, dangerous to rehearse, or potentially possible in the future. Inside a virtual environment, a participant can experiment with high-stakes situations in a low-stakes environment, building intuition and mental models for how the environment reacts. Virtual training environments are prominent in training pilots or power plant operators, for whom live training on real equipment is expensive and at risk for catastrophic mistakes. Game-based training may be high-fidelity recreations of the physical world, but they can also be unscripted, abstract, and open-ended experiences, while still focusing on key aspects of complex tasks.

   The Laboratory has applied training games to several domains, including emergency response to improvised nuclear device detonations and management of delays in air traffic systems during severe weather. A common characteristic of all these games was a focus on only an important slice of the problem rather than on a model of the entire problem space. As a result, the games required just minutes to play a scenario, allowing players to engage in many iterations of a scenario in a single sitting. The Laboratory's methodology that combines repetition of short focused experiences with engagement in longer more detailed experiences has proven an effective approach to cover the full spectrum of complex tasks.
2. Concept exploration and requirement analysis. Predicting what technologies will be useful and impactful prior to building a prototype is error prone and can result in expensive redesigns when operators reject the technology at late stages of a development process. Consultation with experts and end-users is a common approach for gathering functional requirements for future technology. However, this conventional method can be insufficient because experts are often intuitive thinkers who are used to dealing with concrete situations, not abstract thinkers who have a theoretical approach for generalizing knowledge to future scenarios. Moreover, every end user is a novice when thinking about new technologies that may change operational paradigms.

The Laboratory has been using serious games to aid in technology assessment for early-stage R&D prioritization, improved analysis-of-alternatives studies, and development of functional requirements. For example, in remote sensing R&D, the process often starts with an understanding of the phenomenology of the sensing environment and observables of interest, leading to the development of sensor hardware that is then integrated and fielded on the premise that the sensor capabilities are inherently useful. However, many sensor systems have not been jointly developed alongside the decision processes their data are meant to inform. Lincoln Laboratory developed a game to invert this development and acquisition process by starting with an understanding of what information is needed to make decisions and working backward to build an end-to-end workflow that results in actionable information. The gaming process and sensor simulation capabilities were then used to dial in what the technical and performance requirements should be for both the sensors and their data analysis systems.

Additionally, Laboratory researchers also designed games that combine economic game theory with rapid-play digital simulations to collect quantitative data and then crowdsource the ingenuity of human experts. In the game, players select different combinations of conceived capabilities within forced resource constraints, allowing them to formulate and explore different strategies that may deviate from current doctrine and tactics. After players try out the set of capabilities they selected, they get immediate feedback about the utility of these capabilities, build intuition, and iterate to converge on effective combinations of capabilities.

3. Development and evaluation of tools. As candidate technologies move from the requirements process to prototyping, serious games can play a critical role in creating an environment to facilitate purposeful interaction with technology that is not always achieved by feature or user testing. By wrapping the prototype in purpose-built datasets, scenarios, and mechanics, the gameplay pushes users to explore the prototype capability in a rigorous high-fidelity fashion by solving real problems that require informed decisions. Quantitative human-system instrumentation is employed to produce rich interaction data for assessments that drive design improvements, and this process is repeated throughout the development cycle. The Laboratory has used multiple serious games in applying this iterative technique to assess and refine algorithms and workflows aimed at improving multifeed video analysis for counterterrorism and airport security missions.

Areas of Laboratory Innovation

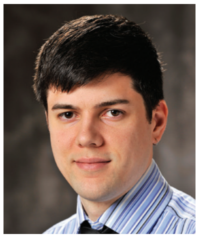
The Laboratory has developed expertise and innovations in key areas of serious game development:

- Scenario and simulation dataset development. Designing a game scenario and the data artifacts that accompany it can be time-consuming and human-intensive. To increase the efficiency of this work, the Laboratory has employed techniques from natural language processing, computer vision, and agent-based modeling to generate synthetic datasets derived from a storyboard and to ground truth real-world datasets, such as news reporting or surveillance video, that are repurposed for gameplay.
- Game mechanics design. Critical to the success of any serious game is its ability to effectively engage users. Laboratory researchers have developed methodologies to design mechanics, scenarios, and underlying simulation behaviors that conform to the user's domain knowledge. Thus, the game earns credibility with and acceptance by users. Resource constraints, scoring rules, and bounds on decisions all require careful consideration to prevent untrustworthy user behavior ("gaming the game").
- Rapid game prototyping. The Laboratory's agile development process enables developers to rapidly examine whether the design choices (e.g., scenarios, allowable player actions, player incentives, underlying models) result in a believable and engaging gameplay experience. The bottleneck in the process is the design stage because designers must be part-time domain experts, experimental designers, psychologists, and data scientists. To address the bottleneck, Laboratory game designers have actively explored creating reusable templates for common game archetypes and leveraging widely available existing game engines.
- Human-subject experiment design. Lincoln Laboratory has spent significant effort researching which factors lead to a robust human-subject experimental design, such as mitigating biases in training approaches, moderators, and hypotheses; limiting the number of experimental variables and options available to players; and balancing the length of play against a data collection opportunity.
- Human performance assessment and decision analysis. The Laboratory's data-driven research methodology and technical framework address game assessment challenges by quantitatively measuring human-human and human-system behavior, rigorously evaluating analytical and cognitive performance, and providing data-driven ways to improve the effectiveness of individuals and teams. This work employs system instrumentation to understand game software and data usage, eye-tracking systems to estimate screen interaction and cognitive load, and wearable sensors to measure team speech dynamics.

Future of Serious Games Research

Engaging and informative games are expected to become part of every preparedness, training, technology development, concept of operation, and operational evaluation process. But that level of penetration requires that the entire design process be fast and inexpensive, and that enough game design automation exists so anyone can be a designer. That end-goal is achievable but will require significant advances in machine learning and artificial intelligence (AI). For example, AI could be used for intelligent individualization of the game progression, with AI examining a player's history for situations that gave the player difficulty and then adjusting scenarios accordingly. Similarly, rather than building a game to operate on a single scenario, developers could use AI to create a game that systematically generates a spectrum of playable scenarios without manual intervention or designer bias. Lastly, while AI can have trouble finding coherent strategies in very large decision spaces, if humans identify strategies worth optimizing, a joint human-AI team could outperform those same humans alone. Serious games are well matched to work through critical issues that face future human-AI systems, such as designing meaningful transparency into what the AI is performing on the user's behalf, and how to earn and calibrate trust in the AI system. The potential reach of serious games has only begun to be explored, and the Laboratory will continue to find unique ways to apply games to the most challenging human-system analysis problems facing the development of future national security sociotechnical systems. ■

About the Authors



Matthew P. Daggett is a member of the technical staff in the Humanitarian Assistance and Disaster Relief Systems Group. He joined Lincoln Laboratory in 2005, and his research focuses on using operations research methodologies and quantitative human-system instrumentation to design and measure the effectiveness of analytic technologies and processes for complex sociotechnical systems. He has expertise in remote sensing optimization, social network analysis, natural-language processing, data visualization, and the study of team dynamics and decision making. He holds a bachelor's degree in electrical engineering from Virginia Polytechnic Institute and State University.

Timothy J. Dasey is the leader of the Informatics and Decision Support Group, which leads programs in homeland security, transportation systems, and biomedical systems, and which provides machine learning and human-machine system analysis skills across Lincoln Laboratory. Previously, he was the leader of the Chemical and Biological Defense Systems Group and a manager in the Weather Sensing Group. He holds a bachelor's degree in electrical and computer engineering from Clarkson University and a doctoral degree in biomedical engineering from Rutgers University.

Adam S. Norige is an associate leader of the Humanitarian Assistance and Disaster Relief Systems Group at Lincoln Laboratory. Currently, he is focused on applying advanced technology to complex disaster relief challenges, with the goal of transforming U.S. disaster relief capabilities. He also teaches innovation in disaster relief at MIT and the U.S. Naval War College. Prior to his work in disaster relief, he led research efforts in advanced sensing architectures for chemical and biological threats, and in serious games for the analysis of complex decision processes. He holds bachelor's and master's degrees in biomedical and computer engineering from Worcester Polytechnic Institute.

Robert M. Seater is a researcher in the Informatics and Decision Support Group at Lincoln Laboratory. He currently works on serious games, requirements analysis, and software engineering. He has applied serious games to a range of topics of interest to the Department of Defense and Department of Homeland Security, including the integration of unmanned aerial vehicles into infantry squads, large-scale emergency response, chemical and biological defense, and naval missile defense. He holds a bachelor's degree in mathematics and computer science from Haverford College and a doctorate from MIT in computer science and requirements engineering.

SIMULATIONS

A Serious Game for Intelligence, Surveillance, and Reconnaissance

A simulation experiment provides hands-on experience analyzing sensor data to discover mobile targets

For the past two years, people at Lincoln Laboratory's Introduction to Intelligence, Surveillance, and Reconnaissance (ISR) Systems and Technology course attended the expected lectures on different sensors and techniques used to provide military operations with ISR data. However, each year's group of about 50 military and civilian government personnel enthusiastically discovered that the course organizers had included a hands-on simulation exercise that demonstrated those sensors and techniques: a serious game that challenged players to use different types of sensors to locate a convoy transporting a mobile missile threat.

In this game scenario, called a red/blue experiment, a simulated (red) threat plays out in a virtual environment while the participants (the blue team) use simulated sensors and tools to make inferences about the threat and to decide upon courses of action. Known as serious because such games are educational tools, the ISR red/blue game was designed to emphasize material covered in the course lectures.

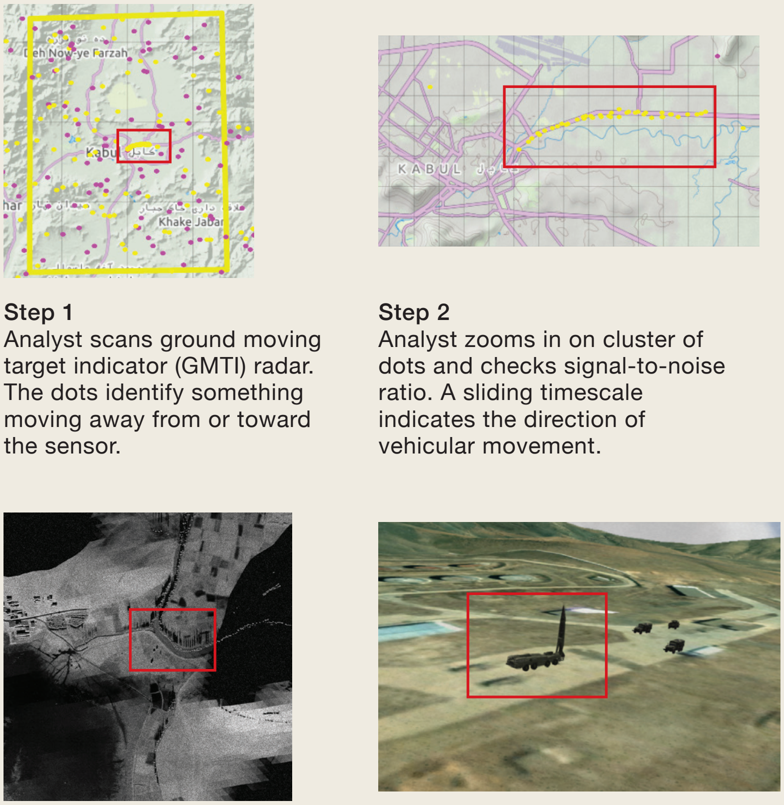
"The game allows the attendees an opportunity to apply the course concepts in a realistic situation and to see firsthand that data exploitation is hard!" said Carol Chiang, a technical staff member of the Laboratory's Intelligence and Decision Technologies Group and a lead developer of the game.

"The game is the outgrowth of many years of technical work in data management and simulation software systems, including many prior red/blue experiments, developed by the group since 2007," said Benjamin Landon, the assistant leader of the Intelligence and Decision Technologies Group. "The game's scenario is driven by the importance of locating mobile targets and the need for rapid decision making in response to identified threats."

During the afternoons of each full day of the two-and-a-half-day course, half the attendees engaged in gameplay while the other half attended seminars and demonstrations. The red/blue game began with a short briefing about the scenario and the tools available to players. The attendees, who were grouped into five teams, had 30 minutes to get acquainted with the tools and 10 minutes to discuss their strategy before they began the first of two 45-minute games. The game scenario, find the mobile target and stop the firing of a missile, was the same for both games, but the second game was complicated by having players contend with decoys and many "confuser" vehicles that were not part of the threat convoy.

"In 2018, we added multiple threat convoys. This addition was to make the game harder and more realistic than the first game that had only one target for the players to find," said Kenneth Mawhinney, another of the game developers. "The players couldn't just focus on monitoring one convoy but had to maintain awareness of a bigger picture."

The game offered players the use of three technologies commonly employed in ISR missions: ground moving target indicator (GMTI) radar, synthetic aperture radar (SAR) imagery, and full-motion video (FMV). Chiang said the recommended utilization of the three modes of data acquisition was a sequence progressing from the use of GMTI flown over a region to determine movement indicative of a convoy, to the use of SAR images for



Step 1
Analyst scans ground moving target indicator (GMTI) radar. The dots identify something moving away from or toward the sensor.

Step 2
Analyst zooms in on cluster of dots and checks signal-to-noise ratio. A sliding timescale indicates the direction of vehicular movement.

Step 3
If vehicles are identified in GMTI data, analyst requests synthetic aperture radar (SAR) imagery for more focused view of vehicles.

Step 4
Analyst reviews full-motion video to positively identify the target.

The work flow depicted here for finding the threat convoy is illustrative of the sequence of tasks that analysts would employ in an actual search for vehicles moving over a broad landscape.

a more focused view of the suspected convoy once it has stopped, and then to an FMV scan to definitively identify one of the vehicles as the one carrying the missile.

Players could choose to task each of the three sensors multiple times. The GMTI radar returns reflected from objects on the ground were plotted on a map of the region; the track from subsequent GMTI sweeps indicated the direction in which the objects were moving. Each request for the SAR produced imagery that helped players refine their view of the objects they had found. The FMV simulation provided the best means of positively identifying the threat convoy, but FMV has a narrow field of view compared to the GMTI and SAR sensors. The mission teams have to use the GMTI and SAR sensors to effectively cue the FMV sensor rather than relying on FMV alone.

On the actual game days, each five-player team was assigned to a different space in which to play the game. Each team was allowed four computer setups. Chiang credits colleague Matthew Daggett with the advice to leave each team with fewer computers than players. “The game is organized around two main analytical tasks—the discovery of information and the integration of that information in order to make decisions. By their nature, the game computers are attractive to use, and everyone wants to see the data. We have found in testing similar serious games that if you give every player a computer, everyone plays the discovery role and no one is integrating information,” Daggett explained. “But if you remove a computer from one person, the team’s only means of ‘discovering’ information is to solicit teammates for information and integrate from that. By having N–1 computers for a team of N, you have the opportunity for a functioning team and not just N ‘analysts’ scanning the data.”

Players were given pregame time to plan a teamwork strategy. They might choose a leader who coordinated the tasking for each sensor mode and then connected the gathered information into an overall picture of the vehicles’ movements. A team could assign one member to each sensing job—searching GMTI data, analyzing SAR imagery, and scanning with the FMV—while the other two members monitored the outputs and kept track of the overall mission. Or, teams could work in pairs or trios to try out each sensor mode while keeping up a dialog about their analyses of the data they were collecting.

During the gameplay, a Lincoln Laboratory staff member was assigned to each team as an advisor. While these helpers followed the game action, they did not assess the effectiveness of a team's organization. However, one player commented that coordination of team roles was key to a team's success.

Development of the ISR game was an intense, two-month effort that required software development and integration, simulation design, and the setup of networks and computers specifically for use by the participants in the ISR course. The Lincoln Laboratory team creating the red/blue experiment was able to draw on software systems, simulations, networks, and displays built for many past projects in the Intelligence and Decision Technologies Group's portfolio. The team employed simulation tools that managed ground and air vehicle routes, interactive simulations, and software that emulated radar and optical sensors based on the underlying virtual world. From an array of components, which included systems for real-time data sharing, sensor simulations, and browser-based map displays and data viewers, the developers assembled a game scenario, tapping into expertise gained from staging red/blue experiments over the past 10 years.

The game is organized around two main analytical tasks—the discovery of information and the integration of that information in order to make decisions.

The most challenging part of developing an engaging red/blue experiment was constructing a scenario that emphasized the technologies and techniques taught in the lectures while still providing a plausible ISR mission thread. Once the sensor modes for the simulation were selected, a scenario for the threat convoy and decoy vehicles was programmed into the simulation software. Next, the Lincoln Laboratory team had to select the level of difficulty for the game, including selecting the amount of simulated cloud cover, haze, and fog that could obscure the optical sensors, and the number of decoy vehicles in the scene. If the game was too easy, it would fail to illustrate the advantages and disadvantages of each sensor mode. Conversely, if the game was too hard, players would quickly lose interest and give up.

The development team had enlisted volunteers to conduct trial runs of the game prior to the course. The iterative trials provided the team with feedback that allowed them to create games that could be managed within the short playing time, but still provide a difficult enough scenario to be challenging. "The level of difficulty was a challenge throughout the dry runs," Chiang said. "Some of the scenarios were too easy. Others were too hard, frustrating players who spent hours playing without finding anything. So rather than having players looking at FMV scans of haze and clouds for 90 percent of their time, we chose to make the game somewhat easy so players could apply the concepts they had just learned and get a result."

Red/Blue experiments are not just demonstrations for students at a course or exercises for military trainees. Researchers can use these experiments to study how humans approach the situations and problems simulated in the scenarios and how they use the new tools, sensors, and automation. Simulation experiments are much less expensive and take less time than fielding a new piece of technology to the military and then asking for feedback. Furthermore, because a red/blue experiment can be repeated in a laboratory setting, researchers can evaluate whether a new tool or analytic has the potential to make a difference to the operational community.

Lincoln Laboratory staff will apply lessons learned from the ISR game players' experiences and feedback to develop more complex scenarios and adapt the tools for use in not only future ISR games but also simulation experiments that may shed light on technologies the Laboratory is investigating. They will also be using the red/blue framework to demonstrate how machine learning techniques can be a benefit for command and control of autonomous systems and for data analysis.

The Role of Serious Games in Ballistic Missile Defense

Brian M. Lewis and John A. Tabaczynski

Serious games have played an important role in the development of ballistic missile defense technology since the mid-1960s. At that time, Lincoln Laboratory initiated a series of games in which researchers assumed the roles of ballistic missile defense (BMD) system operators charged with mitigating a missile attack. Postgame analyses of gameplay led to increased understanding of the technology required to effectively identify and engage missile threats. Throughout the 1970s and 1990s, the BMD community concentrated on developing needed technology, and game playing fell into disuse. With the technology advancements of the 2000s, gameplay re-emerged as an effective way of determining how to exploit the new capabilities against an increasingly sophisticated adversary, and the Laboratory designed games that took advantage of new decision support tools.

The missile defense mission area at Lincoln Laboratory has exploited serious game playing since the early 1960s. The games took many forms and were used by Laboratory researchers to investigate ballistic missile defense (BMD) problems and develop and test solutions. New Laboratory staff, industry personnel, government employees, and warfighters who were invited to participate in the games also got hands-on experience with BMD concepts. Games played an important role during two time periods, separated by about 35 years, and consequently resulted in a wide range of game design and implementation that covered a broad spectrum of objectives. In the intervening period of BMD game-playing inactivity, significant advances in both BMD technology and the computer sciences enabled modern games to achieve a level of sophistication never imagined in the 1960s. Over that same period, the Ballistic Missile Defense System (BMDS) evolved from a simple configuration of radars and interceptors to become a "system of systems."

The Early Years

In the mid-1960s, the U.S. Department of Defense was in the initial stages of developing its first BMD system. Research and field measurement activities were characterizing the physics and observables that would eventually be used by the BMD sensors to identify threatening targets. Large-scale computing systems and the real-time software needed to control a complex BMD system were still in the development stage. The BMD community faced a major question: How does one extract the appropriate information embedded in the sensor data and utilize it in a logical framework capable of successfully engaging an incoming ballistic missile [1]?

During this time, Lincoln Laboratory was the major developer in BMD technology, while Bell Telephone Laboratories was responsible for designing, building, and deploying the actual BMD system. In 1963, Lincoln Laboratory initiated an effort that became known as the Engagement Exercises. The effort brought together Lincoln Laboratory scientists and engineers who were experts in the physics of missile and radar systems, and engineers from the Defense Research Corporation of Santa Barbara, California, who were specialists in the emerging field of missile and defense system computer simulation. These exercises took place prior to the advent of personal computers, and programming was tedious and limited to modest-sized mainframes. The games were played manually by competing teams, housed in separate rooms, relying on pencil and paper and having little or no computer automation available to them. Each game included a defense team, an offense team, and an umpire team. Offense-defense interaction was facilitated by the umpire team, whose members moved between the two competing teams to communicate individual team actions and determine the outcomes of decisions made by each team. (See William Delaney's Looking Back article on page 108 for a personal view of these exercises.)

Each game was preceded by several months of game preparation. The umpire team defined technology and resource constraints. With these constraints in mind, the offense generated weapon inventories and attack strategies, and the defense generated extensive sensor and system architectures, defining their associated measurement capabilities and engagement logic. Strategies were documented on paper with logic diagrams and precalculated decision thresholds.

Once the conflict (game) began, it took several days for the teams to complete the game. After the conflict ended, an extensive period of analysis determined what worked and what needed to be improved. This process generated insight into many facets of the defense system and highlighted technology areas that needed further development. The game was played once or twice a year and grew in sophistication with each cycle. The effort continued for approximately four years.

The Middle Period

For roughly the next 30 years, adversarial games did not play a significant role in the BMD mission area. The major activities within Lincoln Laboratory's program shifted to concentrate on the development of algorithms and the real-time field demonstration of techniques for the critical BMD functions of tracking, discrimination, and decision support. The demonstrations utilized two sophisticated computer systems—one system integrated with the radars at the Kwajalein Missile Range (KMR) in the Marshall Islands and the other located at Lincoln Laboratory. This work went through several iterations, starting with the Lexington Imaging System (LIS) and Kwajalein Imaging System (KIS) effort in the early 1980s and evolved into the Lexington Discrimination System (LDS) and Kwajalein Discrimination System (KDS) by the late 1980s. The LIS and KIS were focused on using state-of-the-art processing hardware to demonstrate the viability of real-time radar image formation. After the capability to produce images in real time was demonstrated, the systems continued to evolve to become the LDS and KDS, which were used to demonstrate a full complement of the critical BMD functions.

Over several years, Lincoln Laboratory conducted demonstrations using the KIS and then the KDS against a variety of realistic ballistic targets at Kwajalein. Prior to implementing the techniques at the KMR sensors, the Lexington system was used to conduct extensive studies, exploiting radar data recorded during live missions at Kwajalein and simulation inputs to make sure the techniques were ready for the live-time field demonstrations. The demonstrations and facilities were important for two reasons. First, they enabled the staff to create a toolbox of real-time software for implementing advanced signal processing and critical BMD algorithms. Second, the demonstrations required the development of the highest-fidelity target models that had been generated up to that time.

As part of the preparation for the field demonstrations, an extensive set of high-fidelity target simulations was developed, along with graphical user interfaces (GUIs), to serve as diagnostic tools for the experimental packages deployed to the field. With the advent of high-throughput computation and advances in high-speed signal processing, these demonstrations were the first in which advanced BMD concepts could

Lincoln Laboratory Simulation Tools

Ballistic Missile Defense Toolbox

These frequently used functions for BMD simulations include modules modeling the physics for ballistic trajectories, torque-free body dynamics, and maneuvering dynamics, as well as utilities for coordinate transforms, mathematical functions, signal processing, and tracking. The toolbox functions were optimized for speed and internally validated.

Lincoln Laboratory LL6D

This 6-degree-of-freedom missile simulation utilizes many of the BMD toolbox functions to create the trajectory files for an entire BMD threat complex. The LL6D emulates unitary boost, staging, object deployments, and individual object dynamics.

Augmented Point-Scattering Model (APSM)

The APSM uses a Lincoln Laboratory radar cross-section signatures-modeling format and a suite of signature interpretation software. Generating intensive scenes on the fly required new techniques because the industry standard signature format, Xpatch, required too much memory and did not, at the time, preserve the phenomenology of interest from pulse to pulse. APSM is based on a point-scattering model [2].

Optical Signatures Code (OSC)

The OSC is a national standard code that generates detailed infrared signatures and that models the output of space-based sensors and interceptor seekers.

Lincoln Laboratory Simulator (LLSIM)

The LLSIM is a simulation framework for generating BMD scenes for all phases of flight and all phenomenology types. LL6D, APSM, and OSC provide the trajectories, RF response for single objects, and infrared response for single objects. LLSIM uses these as inputs to create simulated radar and infrared sensor and data processing output, including multiple-object radar pulses and infrared sensor responses. In addition, discrimination algorithms and decision aids were implemented as part of the data processing. The LLSIM uses an xml file to define the threat and blue force (missile defense) laydown, including the sensors, interceptors, and command and control, and publishes the sensor output to a database for use in visualization software.

Lincoln Laboratory Visualization Interface and Scalable API (LLVISTA)

This visualization software package allows flexible configuration of graphical user interfaces. The tool was developed to decouple the user interface from the BMD scene-generation tools, accomplished via a publish/subscribe implementation. It is able to plot scrolling range-time-intensity, range-Doppler images, and feature/feature plots.

be executed in an autonomous fashion. The GUIs and high-fidelity target and environment models were essential to the success of these field demonstrations. Although the demonstrations were a significant step toward bringing advanced BMD capability to the field, they were limited to single-sensor architectures and remained scripted. A detailed account of these field measurement activities may be found in Chapter 9, Ballistic Missile Defense, of the Lincoln Laboratory history book [2]. The concepts of adaptive sensor architectures, multisensor system design, and centralized/hierarchical system control and decision making were yet to be formalized.

Advances made during this period provided a strong framework for the series of games that would come to fruition in the early 2000s. The core components of this framework are the Lincoln Laboratory 6-degrees-of-freedom (LL6D) trajectory generator, the Augmented Point Scattering Model (APSM), the BMD Toolbox, and the Lincoln Laboratory Visualization Interface and Scalable API (LLVISTA). These enablers were primarily focused on radar sensors.

The Modern BMD Games

In the 1960s, the BMD system employed two radars—a search and acquisition radar and a fire-control radar. These radars did not have much flexibility and handled targets by using a predetermined script with a limited degree of dynamic decision support. Consequently, game playing concentrated on understanding and developing the logic and decision strategy for the battle management functionality.

By the early 2000s, the BMD system concept had evolved, employing multiple radars operating at different frequencies and observing different phases of a threat trajectory. In addition, optical sensors were included as part of the sensing suite. A broad array of algorithms for the critical BMD functions was developed over the intervening years and was extremely sophisticated. Computational power and speeds had reached levels that would allow sensors to operate in a more dynamic and adaptive way. During this time, researchers investigated how to best exploit these new capabilities.

In 2001, adversarial game playing once again became active within the Lincoln Laboratory BMD community under Project Hercules, a national effort sponsored by the Missile Defense Agency (MDA) to advance the state of the art for the critical functions of the BMDS. Figure 1 depicts a notional representation of a generic BMDS. In a simplified view of the modern-day concept, the BMDS consists of a number of individual sensors that gather data about an incoming threat and attempt to identify the lethal target(s). The target state estimates and decision information are passed to a central battle manager where they are integrated with data from all available sensors to provide more complete situational awareness, upgraded fire-control track information, and improved target identification. This information is used to reallocate sensor resources and generate interceptor fire-control solutions for the identified threat targets.

The initial purpose of the modern BMD games was to support the development of sensor algorithms and system architectures that would result in enhanced capabilities for the BMDS. The idea was to understand how the human mind exploited sensor observations to identify the threatening targets while rejecting the accompanying nonlethal targets, and to capture that process so it could be incorporated into decision architectures. The modus operandi was for Lincoln Laboratory subject-matter experts (SMEs), selected from a variety of technical areas, to collaboratively solve specifically designed challenging threat scenarios. The SMEs included the following:

- Data analysts experienced in sensor observables exploitation who could determine relevant and important target characteristics
- Signal processing experts who understood how to extract critical information from sensor observations
- System engineers who understood the resource implications of engagement constraints

To provide motivation, the SMEs were organized into teams that would compete against one another, and trophies were awarded to the winners.

Game Formulation

The new generation of games became known as the red/blue (R/B) exercises. The primary objective of these games was to identify improvements in the sensor

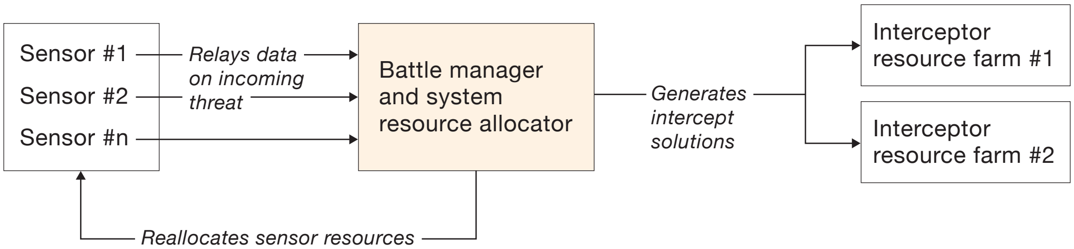

Sensor #1
Sensor #2
Sensor #n
Relays data on incoming threat
Battle manager and system resource allocator
Generates intercept solutions
Interceptor resource farm #1
Interceptor resource farm #2
Reallocates sensor resources

FIGURE 1. In this notional representation of a ballistic missile defense system, individual sensors gather data on an incoming threat and attempt to identify the lethal target(s). The sensor response for each target is passed to the battle manager, which combines data from each sensor to generate more complete situational awareness. This information is used to reallocate sensor resources, generate intercept solutions, and assign interceptors to specific targets.

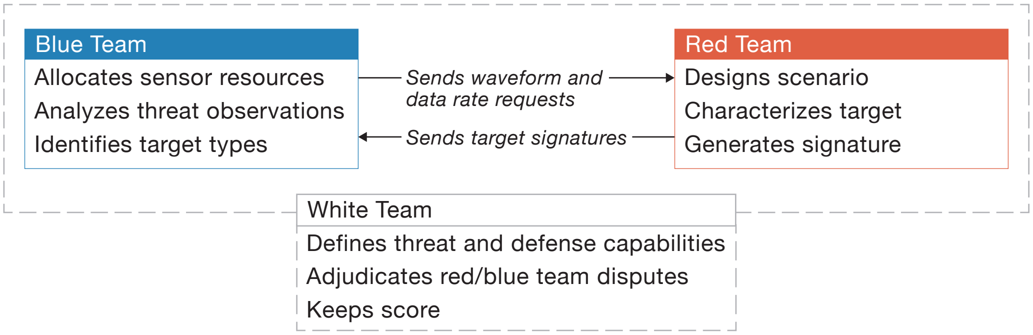

Blue Team
Allocates sensor resources
Analyzes threat observations
Identifies target types
Sends waveform and data rate requests
Sends target signatures
Red Team
Designs scenario
Characterizes target
Generates signature
White Team
Defines threat and defense capabilities
Adjudicates red/blue team disputes
Keeps score

FIGURE 2. The team structure of a ballistic missile defense game includes red (offense), white (umpire), and blue (defense) teams.

decision architecture. Figure 2 provides an overview of the organization of the initial games and shows the responsibilities of the teams and the primary interactions between the teams. Several blue teams competed against each other to mitigate a common threat generated by the red team. The white team, similar to the old umpire team, postulated a BMD problem, and the red team had several months to define the threat and generate the sensor observables. The red team was composed of Lincoln Laboratory staff who worked with the MDA's threat engineering team, the intelligence community, and the Laboratory's Engineering Division. The red team reviewed the known offensive capabilities of peer and rogue nations, and worked diligently to ensure that any threat components incorporated into the game reflected the engineering capabilities of an actual adversary.

The generation of very high-fidelity simulation sensor observables was the most important and tedious part of the game. This task was critical because the games were intended to challenge the SMEs' ability to discriminate the targets on the basis of sensor observations. For the development process to have credibility, the information contained in the sensor signatures had to be as realistic as possible. The years of modeling experience obtained during the 1980s and 1990s in support of the Lexington Discrimination System development were critical to making the BMD games realistic and able to contribute to the development of BMD technology.

Prior to the game, the white team defined a scoring structure so that prizes could be awarded to the winning team. The game was played over two days and included a preparation phase and a postgame analysis phase. During the preparation phase, each blue team learned the capabilities of its sensor and the nature of the threat and defense problem. On the first morning, they were guided through a simple version of the game that contained no countermeasures. The blue team spent the afternoon developing its strategy for sensor measurement and data exploitation, and documenting the strategy with flow charts and decision graphs.

On game day, the red team was allowed to utilize countermeasures. Each blue team occupied a separate room in which a white team observer recorded the team's play. Once the engagement was underway, the blue team was permitted to make procedural modifications, and the white team documented the changes accordingly. During the postgame phase, the white team identified the strengths and weaknesses of the blue team's methodology in order to develop a better understanding of how to improve the sensor decision architecture. In addition, participants made valuable recommendations for algorithm upgrades and graphical user interface (GUI) improvements.

The Evolution of the BMD Game

The timeline continuing along the bottoms of the following pages provides a history of game development and highlights key features in the game's evolution. Several significant transitions in the level of game sophistication are described in the following text.

2001

DATE: 10/31

DESIGNATION: Red/Blue (R/B) 1

SENSOR SUITE: Midcourse (MC) radar

PURPOSE: Established best use of radar data to identify targets and seek discrimination ideas and features for red threat of interest

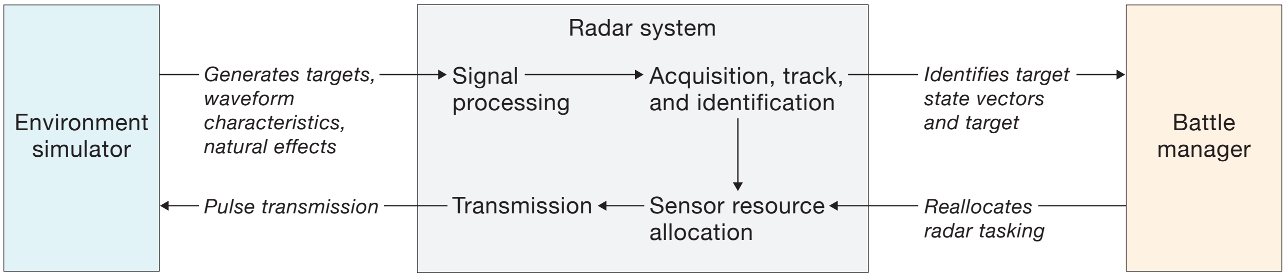

Environment simulator
Generates targets, waveform characteristics, natural effects
Pulse transmission
Radar system
Signal processing
Acquisition, track, and identification
Transmission
Sensor resource allocation
Identifies target state vectors and target
Reallocates radar tasking
Battle manager

FIGURE 3. This notional representation of a ballistic missile defense sensor depicts the interplay between the environment simulator, the radar system, and the battle manager. The battle manager provides tasking to the radar, which allocates the resources (schedules the pulse-repetition frequency and desired waveform for each object) and then transmits the pulses. The simulator returns the multi-object threat complex response for each transmitted pulse to the signal processor, which compresses the pulse and adds the proper system noise. If objects are above the noise threshold, the detections are associated with existing object tracks. Long-term tracks receive target identification. This information is sent to the battle manager and may be used for the next resource period.

Early Radar Games (Red/Blue [R/B] 1, 2, 3, and 7)

The objective for the initial set of modern games was to understand the strategies that each team employed, with the intent of integrating the successful sections of the logic and processes into a decision architecture for a computer-controlled radar. The focus was to specify the data needed and decision logic required to correctly classify an observed target.

Figure 3 depicts a notional sensor configuration. The game software processed signal information from the target scenario and placed objects into track—an estimated trajectory based on the object's associated detections in range, azimuth, and elevation relative to the radar's boresight. The blue team managed the radar resources to collect the data necessary to support their decision architecture. The R/B 7 provided the software structure to integrate several of the advanced algorithms into a comprehensive software package. This package enabled the blue team to schedule algorithms so that the outputs could be used by the architecture. Equally important, R/B 7 allowed the white team to examine a variety of ways to exploit new concepts and evolve a best-practice approach.

Interceptor Games (R/B 4 and 6)

In late 2002, the game (R/B 4) still emphasized a single sensor but addressed a different sensing phenomenology by considering a multiband infrared (IR) sensor aboard a missile defense interceptor that targeted a multi-object threat complex. However, the interceptor was a moving platform with a limited field of view and the ability to divert to a given object in the threat complex. As the interceptor approached the complex, objects dropped out of the interceptor's field of view and escaped the interceptor's reach. To ensure adequate viewing time, the blue team managed the sensor field of view, the interceptor's approach vector to the complex, and containment for objects of interest. The R/B 4 was the first of two games focused solely on the IR seeker. In late 2004, R/B 6 included a visible-band optical sensor that allowed the blue teams to explore the utility of this additional capability.

2002

»

DATE: 02/14
DESIGNATION: R/B 2
SENSOR SUITE: MC radar
PURPOSE: Established best use of radar data to identify targets with long viewing time

DATE: 07/17
DESIGNATION: R/B 3
SENSOR SUITE: Forward-based (FB) radar
PURPOSE: Established best use of radar resources for a forward-deployed radar with limited viewing time

DATE: 10/27
DESIGNATION: R/B 4
SENSOR SUITE: MC infrared (IR)
PURPOSE: Used IR data for an intercontinental ballistic missile–class interceptor

System-Level Games (R/B 5, 8, 9, and 10)

The first multiple-sensor game was R/B 5, which was conducted in spring 2003. This game featured two radars, each with different sensor attributes (e.g., sensitivity, frequency) and located to allow the radars to observe the threat with different viewing geometries. The a priori sensor positioning provided the defense with a much richer set of observables and allowed for more sophistication in the decision support design than previous games. In this scenario, the blue teams did not contend with inter-sensor bias, i.e., slight imperfections in sensor pointing that could interfere with sensor-to-sensor object correlation.

The real-world challenges of sensor bias and correlating (mapping) objects from one sensor to another were introduced in 2005 with R/B 8. The blue teams faced a complex scenario for discrimination and a new challenge of imperfect sensor-to-sensor handover from forward-based radar to midcourse radar and from the midcourse radar to the interceptor. Along with choosing waveforms to help discriminate between objects, the blue teams adjusted the tracking resources (between none, low-resource, and high-resource track waveforms) on each object in the threat complex to help resolve correlation ambiguities.

In early 2007, the next major evolution of the game (R/B 10) included a multi-threat raid scenario. The red team devised five threats of varying complexity. The game was played in scaled real time, and additional decision aids were provided, including a prototype decision architecture and a more elaborate fire-control display than was used in previous games. The software provided estimates for object lethality and decision confidence while the fire-control display included interceptor availability and a dynamic interceptor-scheduling GUI. The white team observed how the blue team utilized the architecture output, concentrating on the human-machine interaction and the use of decision confidence measures.

In parallel to the large two-day version of the R/B games, the developers produced smaller-scale, one-hour games (mini-R/B or MRB) to play at various missile defense conferences, workshops, and courses. These venues included the Ballistic Missile Defense Joint Advisory Committee Meeting, later called the Air and Missile Defense Technology Workshop (AMDT), the Lincoln Laboratory BMD Technology course, the Missile Defense Sensors, Environments, and Architectures Conference (MD-SEA), and the National Fire Control Symposium. The first three of these mini-games, MRB 1–3, were scaled-down versions of full games.

MRB 4 introduced a new era in R/B small-scale games. It included an emulation of MDA's newest proposed system architecture. The portable game was used to educate participants about the BMD system's operation and to study human interaction with the proposed architecture.

In May 2009, MRB 5 was introduced and included an update that allowed blue teams to assign specific roles and functions to individual team members to more realistically reflect missile defense system operation. The Lexington Decision Support Center provided separate control rooms for the functional subteams of each blue team to perform their roles during gameplay. The software was updated to pipe the same threat information to each room, which displayed an emulated sensor or system function output. For each blue team, two to three members served as the operators for a forward-based radar, two to three members served as the operators for a midcourse sensor, and two to three served as the command, control, battle management, and communications (C2BMC)/ground-based midcourse defense fire-control operators. In addition, each sensor

2003

DATE: 05/08
DESIGNATION: R/B 5
SENSOR SUITE: FB and MC radars
PURPOSE: Performed system-level discrimination for multiple radars with perfect target handover between radars

2004

DATE: 03/17
DESIGNATION: R/B 6
SENSOR SUITE: MC IR and visible radars
PURPOSE: Investigated utility of visible data for discrimination

DATE: 10/22
DESIGNATION: R/B 7
SENSOR SUITE: MC radar
PURPOSE: Integrated advanced algorithms into R/B framework and observed analyst utilization of algorithms and approaches to schedule waveforms for input into algorithms

team communicated with the fire-control operators via text and graphical communications and used voice links between the rooms for discussion.

By May 2010, this distributed approach to the operations was enhanced to include two airborne infrared (ABIR) unmanned aerial vehicles that fed data into the C2BMC node. This game focused on sensor resource management, in particular using the two ABIR vehicles for tracking in a raid environment. The goal was to provide sufficient track quality to engage the maximal number of threats in a raid. For AMDT 2011, the ABIR element was augmented with a discrimination capability.

The BMD Games Infrastructure

The initial version of the modern BMD games was quite modest, residing on a small network of laptop machines. It exploited many of the target signature simulation and discrimination tools that had been developed in support of Lincoln Laboratory's long-standing BMD discrimination technology program. Many of these software packages were first used in the Lexington Discrimination System and evolved in quality with each field experiment.

As the games evolved, they incorporated more sophisticated threats, sensor processing algorithms, and decision support tools, and eventually required a larger network of computing hardware to accommodate gameplay needs. In a parallel effort, Lincoln Laboratory was developing a BMD Decision Support Laboratory that exploited the capabilities of the Laboratory's high-performance computing facility [2]. This facility, known as the Lexington Decision Support Center (LDSC), was the culmination of a multidecade evolution of Lincoln Laboratory simulation tools that were developed in support of discrimination technology. The LDSC consisted of several separate, but highly integrated, laboratories. One laboratory was dedicated to the development of very high-fidelity sensor and environment simulations. A second laboratory was dedicated to multisensor information fusion and battle management, while a third housed the development and testing of decision support tools for BMD.

In May 2009, a distributed defense system game was developed for the Lincoln Laboratory Joint Advisory Committee meeting and was installed in the new LDSC facility. The advantage of this instantiation was that it allowed a team to be broken into subteams and placed in separate rooms with specific displays for the team's sensor control, data fusion and battle management, and weapon-control functions. The displays were linked by voice communication in a manner similar to the way a distributed weapon system would be implemented. This arrangement allowed for the development of additional interactive displays that addressed how the separate subteams could communicate efficiently.

2005

DATE: 05/17
DESIGNATION: MRB 1
SENSOR SUITE: MC radar
PURPOSE: Scaled down the version (both in time and complexity) of R/B 7; first game at Ballistic Missile Defense Joint Advisory Committee (BMD JAC). Used during BMD technology courses hosted at Lincoln Laboratory

DATE: 12/7
DESIGNATION: R/B 8
SENSOR SUITE: FB and MC radars and exoatmospheric kill vehicle (EKV)
PURPOSE: Introduced complexity into the game with blue teams performing radio-frequency (RF)-to-RF handover and RF-to-IR handover. Added low pulse-repetition frequency (PRF) and high PRF track waveforms to allow for more handover control. Added user interface for correlation and sensor bias removal

System Resource Allocation Games (AMDT 2012)

By 2012, the game had changed radically. Its focus had shifted from the details of sensor data exploitation to the investigation of high-level system issues, such as preplanned disposition of assets and the real-time allocation of defense resources during battle. The resource allocation game introduced a number of new features and was the first game that combined air and missile defense. It was also the first game in which a red team played interactively against a blue team and in which random events were used to model the fog of war, i.e., uncertainty in situational awareness experienced by participants in military operations. Since sensor data exploitation was no longer an objective to be explored in these games, no attempt was made to model the signatures of the various targets or to model various decision algorithms.

Game Play

The actual game-playing experience has changed significantly during the game's history. Early versions employed projected displays and allowed the clock to be stopped for team discussions. By 2007, R/B 10 featured a reduced tempo clock and an uninterrupted timeline. At the game's most mature stage, individual interactive desktop displays portrayed information unique for each operator position.

Figure 4 shows a blue team on game day. The game control operator sits at the console at the left. The right screen displays selection options for the radar resources. On the left and center screens are wideband radar displays that depict radar returns from several targets. The blue team analyzes this information to determine the team's future moves.

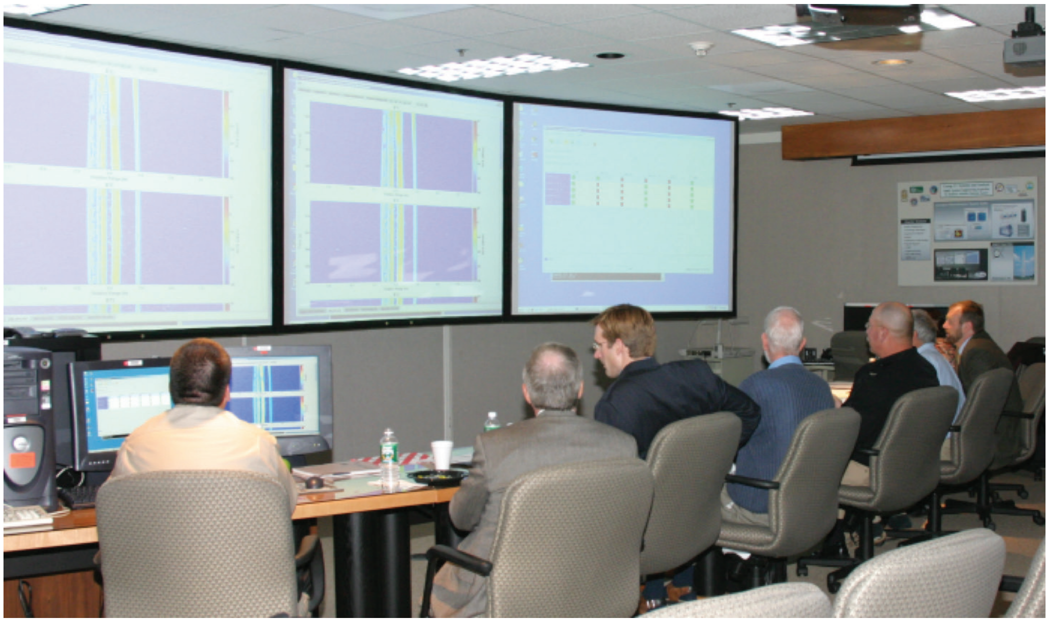


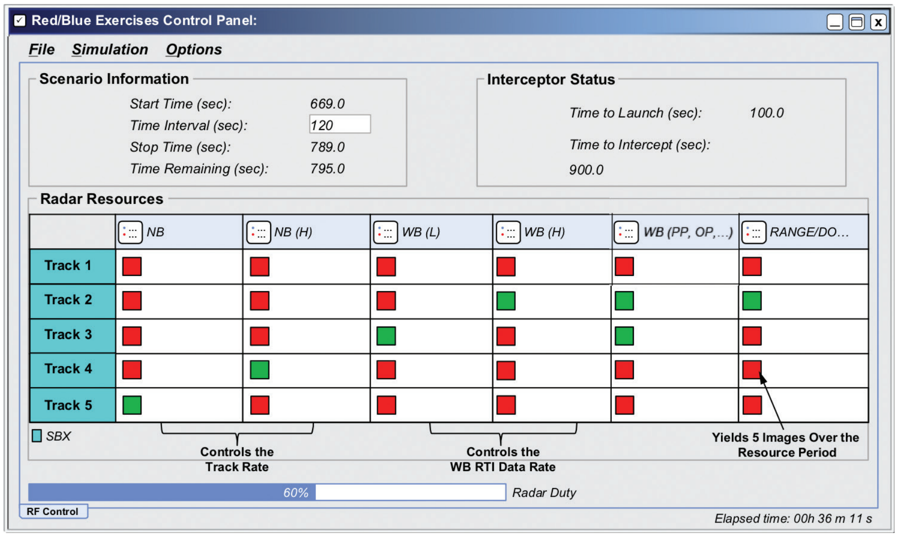
FIGURE 4. The blue team analyzes target observations, assesses engagement status, and prepares radar resource requests for the next time interval.

2006

DATE: 05/03
DESIGNATION: R/B 9
SENSOR SUITE: FB and MC radars
PURPOSE: Added impact-point prediction

DATE: 05/22
DESIGNATION: MRB 2
SENSOR SUITE: FB and MC radars
PURPOSE: Scaled down the version of R/B 8; presented at BMD JAC 2006

DATE: 10/25
DESIGNATION: MRB 3
SENSOR SUITE: FB and SM-3 radars
PURPOSE: Introduced a regional scenario with compressed sensor and playing timelines. Played at Missile Defense Sensors, Environments, and Architectures Conference, at the MDA for the program office, at the BMD JAC, and in Huntsville, Alabama. More than 100 participants



☑ Red/Blue Exercises Control Panel:

File Simulation Options

Scenario Information

Start Time (sec):	669.0
Time Interval (sec):	120
Stop Time (sec):	789.0
Time Remaining (sec):	795.0

Interceptor Status

Time to Launch (sec): 100.0

Time to Intercept (sec): 900.0

Radar Resources

	NB	NB (H)	WB (L)	WB (H)	WB (PP, OP,...)	RANGE/DO...
Track 1	🟥	🟥	🟥	🟥	🟥	🟥
Track 2	🟥	🟥	🟥	🟩	🟩	🟩
Track 3	🟥	🟥	🟩	🟥	🟩	🟥
Track 4	🟥	🟩	🟥	🟥	🟥	🟥
Track 5	🟩	🟥	🟥	🟥	🟥	🟥

SBX
Controls the Track Rate
Controls the WB RTI Data Rate
Yields 5 Images Over the Resource Period
60%
Radar Duty
RF Control

Elapsed time: 00h 36 m 11 s

FIGURE 5. The radar control panel displays current status and waveform option buttons, including narrowband (NB) low-resolution waveform, wideband (WB) waveform, low pulse-repetition frequency (PRF) transmission (L), and high PRF transmission (H). Each option results in different levels of information quality. Green indicates that the sensor is going to employ that waveform on the given track, and red indicates that the sensor is not using the waveform on the given track.

Figure 5 depicts the GUI used to manage the resources of a generic long-range, wideband radar. Each choice results in a different fraction of radar resource devoted to the selected function. Typically, higher radar resource allocation provides improved levels of information quality. The trade-off between resource allocation and information quality is established by the choice of sensor technology assumed during the game design phase. The blue team then schedules when to collect the data on the objects in track and decides how to optimize information gain under current resource constraints. The green and red toggle boxes indicate how the blue team opted to schedule and collect data on the targets in track. The left-hand column identifies the track file, and the row shows the resources the team assigned to that particular target. In this case, Track 2 represents a target that is in track, and the team opted to gather the highest-quality wideband discrimination data that the radar is capable of collecting. At the bottom of the control panel, the Radar Duty bar indicates the fraction of total radar resource being consumed by all tasks currently executing, and shows the radar to be operating at slightly more than half its full capacity.

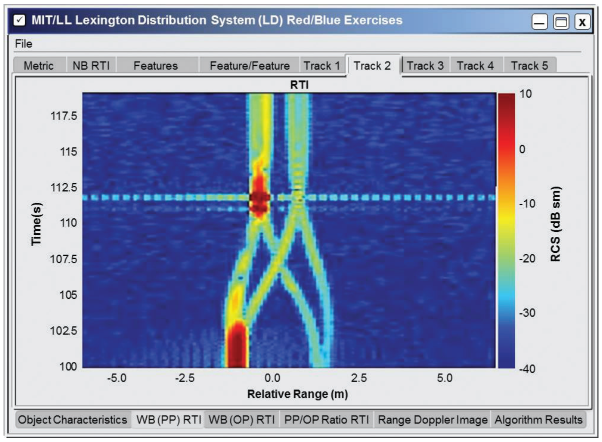


FIGURE 6. This display shows a wideband radar range-time-intensity plot for Track 2.

Figure 6 displays a range-time-intensity (RTI) plot for a target being tracked by the radar. At the top of the

2007

DATE: 04/18

DESIGNATION: R/B 10

SENSOR SUITE: FB and MC radars, fire control

PURPOSE: Premiered scaled-down real-time games and more elaborate fire control with inventory and multiple weapon sites. Unveiled first raid scenario, featuring automation of decision logic and correlation to investigate interaction of humans with decision aids

2008

DATE: 05/15

DESIGNATION: MRB 4

SENSOR SUITE: FB and MC radars, fire control

PURPOSE: Scaled down the R/B 10 version to teach newly adopted missile defense object-targeting concepts

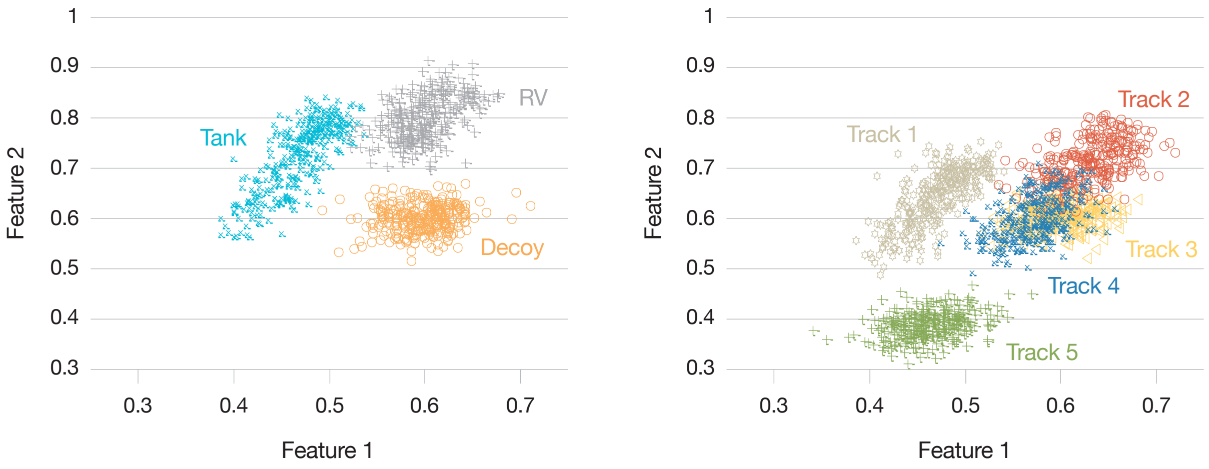


FIGURE 7. A feature/feature plot shows a comparison of extracted features for each object in a scene. In this example, the left panel displays labeled a priori training information, and the right panel displays game-day information from objects in track with unknown types (marker colors are used to indicate that the features are from the same tracked object; colors are randomly assigned).

screen, the team can select from several tabs to examine particular plots of the data collected by the sensor. The first four tabs provide information for the entire sensor collection. The Metric tab provides altitude versus time; the narrowband low-resolution waveform, NB RTI, tab has the radar cross-section (RCS) response for the collection of objects in the scene in range over time; and the Feature/Feature tab provides a comparison of the extracted features (such as depicted in Figure 7) for each object in the scene. These plots are updated in real time with data from the radar's scheduled waveforms.

The remaining tabs exhibit object-specific output. In the example given, there are five tracks, and the tab for Track 2 is selected. There are six additional tabs along the bottom of the GUI to display data collected with the suite of waveforms. In this example, the wideband RTI is selected, and the collected radar response is shown. There is a tab for each of the other waveforms and an additional tab for algorithm results. The Features tab includes a dropdown menu of the different discrimination features derived from the collected sensor objects. The available features were based on legacy BMD features and new prototype features derived from Project Hercules or previous R/B games.

Figure 7 depicts a feature/feature plot. The left panel shows the a priori data from a training day scenario while the right panel shows the output from the game-day scenario. The blue teams can select features in real time for the x- and y-axes to explore feature combinations that provide the greatest decision-making utility.

As expected, the a priori data on the left does not match the game-day observations on the right. The blue

2009

DATE: 05/19

DESIGNATION: Joint Advisory Committee (JAC) meeting, 2009

SENSOR SUITE: FB and MC radars, fire control

PURPOSE: First distributed game; blue teams divided and operated different sensors and fire control

2010

DATE: 05/18

DESIGNATION: JAC 2010

SENSOR SUITE: Airborne infrared (ABIR) radar

PURPOSE: Introduced distributed, sensor resource management for two ABIR platforms and Ground-Based Midcourse Defense fire control with no discrimination

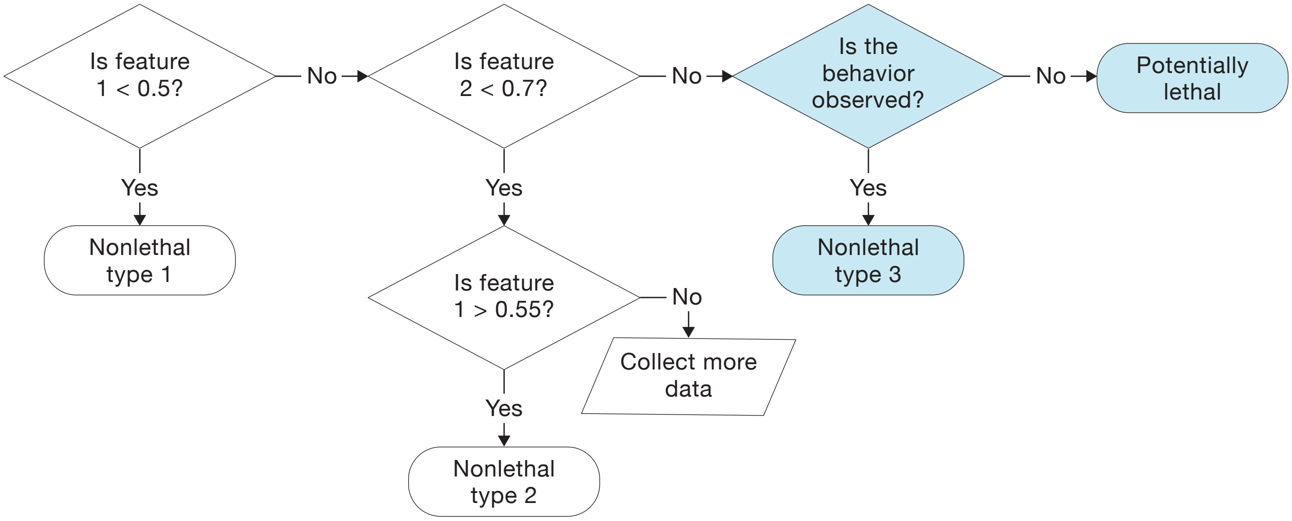


FIGURE 8. This chart represents a portion of a notional blue team discrimination architecture. The blue team used the data provided on training day to identify feature thresholds and determine lethality. During the actual game day, there were off-nominal conditions (i.e., operational or environmental factors were not as planned), and a new feature was used to break a tie from multiple identified lethal objects.

teams must decide how to manage the measurement resources applied to the different objects in order to resolve any uncertainty or ambiguity, and they must interpret the changes in the appearance of the objects that they expected from prior experience. The blue teams faced several questions: Did the red team disguise the reentry vehicle? Was there a deployment malfunction? Are there countermeasures? Are there multiple reentry vehicles? The blue teams could rely on their discrimination architecture logic to request additional sensor resources to resolve the uncertainties.

Figure 8 represents a portion of a discrimination architecture developed in response to the training-day experience. The blue shapes were updated with game-day innovations.

The white team observers kept detailed notes about how each blue team executed its strategy and adapted its decision architecture. To decide which blue team had won, the white team used an overall metric based on lethal objects correctly engaged. If a tie occurred, the white team used tiebreaker metrics, such as the number of objects correctly discriminated, the number of interceptors employed, and resources and time efficiently used. The final out-briefing included a short strategy discussion from each blue team, and the winners were awarded trophies.

Outreach

Over time, the scope of the games broadened to include all aspects of the BMD system of systems. This expansion allowed for the investigation of a wide range of

2011

DATE: 05/17

DESIGNATION: Air and Missile Defense Technology (AMDT) Workshop 2011

SENSOR SUITE: ABIR radar

PURPOSE: Distributed and controlled two ABIR platforms, performed discrimination and passed information to ground fusion center

2012

DATE: 05/15

DESIGNATION: AMDT 2012

SENSOR SUITE: System level

PURPOSE: Eliminated deliberative, planned red scenario. First game with red team being played by game participants and first integrated air and missile defense game. Red team attacks and blue team defends high-value assets (e.g., carrier)

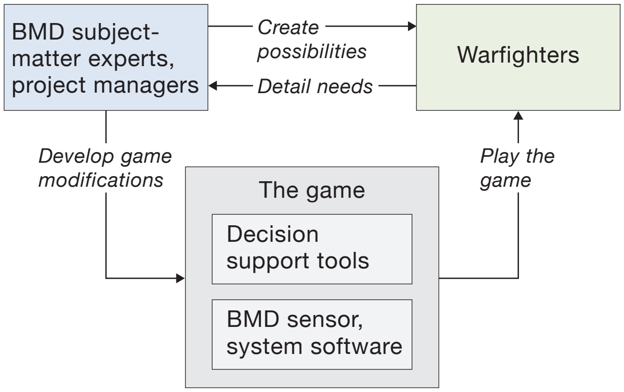

BMD subject-matter experts, project managers
Create possibilities
Detail needs
Warfighters
Develop game modifications
Play the game
The game
Decision support tools
BMD sensor, system software

FIGURE 9. The gaming process facilitates interaction between the development community and the warfighter. This graph represents important aspects of interaction between the developer and the user, such as warfighter feedback influencing decision support tools.

potential capability improvements. Figure 9 depicts the important aspects of the interaction between developers and users, and the influence of this intereaction on system technology. The body of blue team participants eventually expanded to include expert BMD analysts and program managers from Lincoln Laboratory, the prime contractors, and the MDA. The warfighter was also brought in during the later phases of game development to help the developers understand not only the challenges faced by the military system operators but also potential future system improvements. The participants provided welcome feedback regarding the GUIs that the Laboratory was developing to display the threat information and decision aids.

In later years, the R/B game was used as an educational tool. The introductory material was transformed into a tutorial on BMD discrimination, and the scenarios were used to enhance understanding of the adversary's capability and the potential of new technology to mitigate the evolving threat. Such games were used in the Lincoln Laboratory BMD Technology Course and played at MD-SEA conferences, the American Institute of Aeronautics and Astronautics BMD conferences, and Lincoln Laboratory's annual Air, Missile, and Maritime Defense Technology Workshop. Participants included employees from MDA, researchers from federally funded research and development centers, warfighters, and prime contractors. At some events, participation exceeded 100 individuals. Graphical user interfaces and decision aids were updated for each subsequent game, and the game focus evolved to address MDA's most pressing issues.

Further Development

As the game was exposed to a broader community, the U.S. Navy took particular interest in its further development. In 2013, the Office of Naval Research established a project to evolve the game into a training tool for Navy operators. The Laboratory and a commercial gaming company, Pipeworks, converted the technology to the standards required for fleet training operations. A detailed discussion of the effort is provided in an article titled "Strike Group Defender" on page 25. Other mission areas at Lincoln Laboratory recognized the advantages of using gameplay to develop and test sensing and decision support technology. An early adopter was the Laboratory's intelligence, surveillance, and reconnaissance program, which developed games specific to that mission area.

The serious games concept and underlying software structures continue to be used in several technical areas. The detailed simulation tools that support algorithm development and the BMD games are still relevant and

2013

DATE: 05/16
DESIGNATION: AMDT 2013
SENSOR SUITE: Electronic warfare (EW)
PURPOSE: Created similar attributes to AMDT 2012 version, with blue team using soft-kill techniques against a red team cruise missile attack

2014

DATE: 06/04
DESIGNATION: AMDT 2014
SENSOR SUITE: EW
PURPOSE: Established game company version of the 2012 game

are continually updated for applications in the various system studies conducted in Lincoln Laboratory's BMD mission area. As the BMDS matures and increases in complexity, it can be anticipated that a new round of BMD games will emerge. ■

References

1. S.A. Hildreth, "Ballistic Missile Defense: Historical Overview," Congressional Research Service Report for Congress. Washington, D.C.: Library of Congress, 2007.
2. A.A. Grometstein, *Technology in Support of National Security*. Lexington, Mass.: MIT Lincoln Laboratory, 2011.

About the Authors

Brian M. Lewis is the leader of the Advanced Undersea Systems and Technology Group at Lincoln Laboratory. He is responsible for managing research and development for a broad range of undersea assessments and technology development efforts. He joined the Laboratory in 2003 as a staff member in the Advanced Concepts and Technology Group and developed advanced radar discrimination algorithms and high-fidelity sensor emulations. In 2008, he moved to the Missile Defense Elements Group and focused on systems analysis and modeling and simulation in the Verification, Validation, and Accreditation program for the MDA's BMDS Sensors Directorate. He has also served as the assistant and associate leader of the BMDS Integration Group, leading the Laboratory's technology development efforts for the Aegis BMD System. He holds a bachelor's degree in mathematics with a concentration in computing from Morehead State University and master's and doctoral degrees in applied mathematics from North Carolina State University. During graduate school, he was employed as a research assistant at Los Alamos National Laboratory and as a member of the technical staff at the Aerospace Corporation.

John A. Tabaczynski joined the technical staff at Lincoln Laboratory in November 1966 after working at NASA's Jet Propulsion Laboratory. Having served as assistant and associate group leader, he became the leader of the BMD Analysis and Systems Group in 1978. In 1984, he was promoted to associate head of the Ballistic Missile Defense Division. His primary technical interests have evolved over the years. Initial tasks involved the application of Kalman filtering techniques to the problem of radar tracking and discrimination. Through the 1970s, he participated in major defense system studies that shaped the evolving BMDS architecture. During this period, he was involved in defining the technology and specifications for the U.S. Army's family of X-band radars. As an outgrowth of this activity, he initiated Laboratory activities in support of the MDA's Sensor Technology Program. In the late 1980s, as manager of the Laboratory's Kwajalein Missile Range Program, he was responsible for conducting the real-time demonstration of BMD technology and algorithms on live missions into Kwajalein. Through the 1990s, he promoted advanced radar technology for BMD applications and co-led an Army BMD study to define the next-generation missile defense radar. This study informed the Laboratory's current program in over-the-horizon radar. In 2000, he worked with staff at the MDA to stand up and execute Project Hercules, a national effort in the development and testing of ballistic missile discrimination technology. In 2004, he was appointed a principal laboratory researcher and continued to work on areas of advanced radar technology. He is currently writing a textbook on advanced radar signal processing for imaging and radar signature modeling. He holds a bachelor's degree from MIT and master's and doctoral degrees in electrical engineering from Purdue University.

2015

DATE: 04/07

DESIGNATION: AMDT 2015

SENSOR SUITE: System level

PURPOSE: Added positioned multiple radars to optimize performance in a raid scenario

2016

DATE: 05/17

DESIGNATION: Air, Missile, and Maritime Defense Technology Workshop 2016

SENSOR SUITE: System level

PURPOSE: Introduced undersea component into 2012 infrastructure

Strike Group Defender

G. Mark Jones, Matthew C. Gombolay, Reed E. Jensen, and Steven L. Nelson

Defending U.S. Navy ships from the growing danger presented by modern anti-ship cruise missiles is a formidable challenge. Lincoln Laboratory, partnering with government and industry, developed the game-based trainer *Strike Group Defender* to equip the modern sailor with the knowledge and skills necessary to address the evolving threat. The combination of the immersive interface with novel machine learning and artificial intelligence techniques is advancing the state of the art in interactive training.

From extending the reach of the military, to countering piracy, to defending against ballistic missiles, to responding to natural disasters, the surface fleet of the U.S. Navy performs a range of critical missions to protect national interests at home and abroad. To provide the diverse capabilities required by these missions, the Navy has fielded a fleet of more than 100 major surface combatants, ranging from versatile littoral combat ships and guided-missile cruisers to the immense nuclear-powered aircraft carriers [1]. The total crew size represented by these ships is well in excess of 90,000 sailors, underscoring their importance as a major Navy asset [2].

Though the fleet is an undeniably formidable global force, potential adversaries are developing advanced weapons, intent on putting U.S. ships and their crews at ever-increasing risk [3]. The emerging capabilities and proliferation of modern anti-ship cruise missiles (ASCMs) present a considerable threat to the surface ships of the Navy and their missions [4].

In recognition of this evolving threat, the Navy has a wide array of counter-ASCM systems, both deployed and in development, with the goal of equipping each ship (Figure 1) with a robust layered defense [5]. Each countermeasure system provides unique and complementary capabilities that must be employed quickly, correctly, and judiciously to mitigate the ASCM threat.

While the diversity in defensive systems is designed to enhance robustness for addressing the wide variety of ASCM types, the additional complexity of the combined defensive system presents a significant challenge for the sailors tasked with responding to ASCMs. For example,

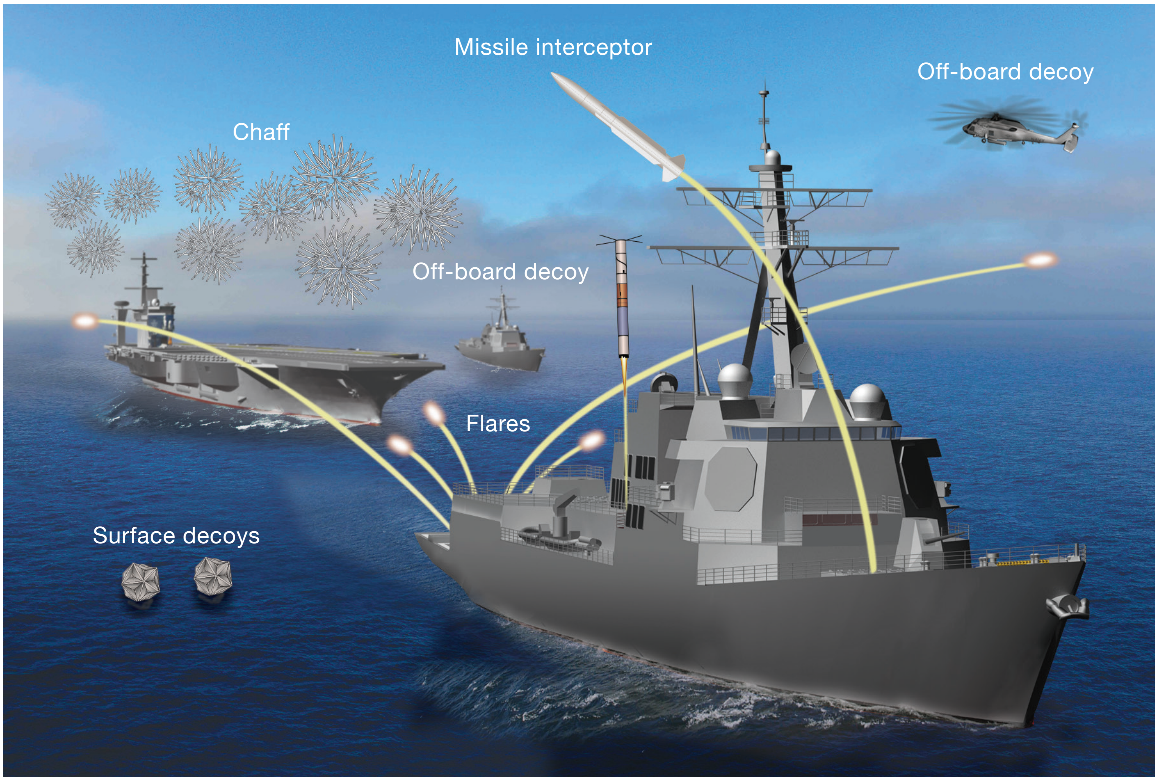


FIGURE 1. The various systems depicted in the illustration can be deployed to counter anti-ship cruise missiles.

assigning five countermeasures against five threats allows more than 100 distinct combinations of countermeasures, though only a few of these potential choices may actually result in a positive outcome for the defense. More realistic situations than this simple example have greater numbers and types of both attacking missiles and countermeasure systems, with additional complications from timing and geometric deployment considerations, and are therefore exponentially more complex. The challenge for the modern sailor is to select the correct course of defensive action, often on very short timelines and with incomplete information, from the large number of choices afforded by the array of countermeasure systems.

A critical factor in preparing the modern sailor to address complex ASCM scenarios is clear, accurate, and detailed training [6]. In recognition of this, Chief of Naval Operations Admiral John Richardson has made one of his four principal thrusts for the Navy to "achieve high-velocity learning at every level." In particular, he suggests that the Navy "expand the use of learning-centered technologies, simulators, online gaming, analytics and other tools as a means to bring in creativity, operational agility, and insight" [7].

The serious game *Strike Group Defender* (*SGD* for short) was designed with this training need in mind, harnessing the immersive nature of modern video game technology, coupled with cutting-edge adaptive machine learning techniques, to provide the Navy with a flexible training and evaluation tool suitable for addressing demanding, realistic modern scenarios. In the end, the goal of *SGD* is to enable sailors to better defend themselves and their ships against the real dangers they face in their naval assignments.

Why a Video Game?

Because the purpose for the vast majority of video games ever produced has undeniably been entertainment, there has been a natural uncertainty and guardedness about games' effectiveness and legitimacy for educational uses [8–10]. Even so, familiar schoolhouse games, such as *The Oregon Trail*, have occupied a niche market in the gaming world since the 1980s [11]. As technology has improved and the proliferation of video games into everyday life has increased, the interest in using video games for education also has grown [12]. The development of *SGD* as a video game was driven by several key factors: the clear connection games provide to young people, the game industry's development of supporting technology, and the natural representation of defense against ASCMs as a two-sided game.

Connection

Part of the growing interest in leveraging video games for instruction flows naturally from the realization that today's students have never known a world without the influence of video games [13]. To put this in perspective for the Navy, where the average enlisted crew member is 22 years old, the original *Mario Bros.*™ game was a quarter-century old when today's sailors were in middle school and the venerable progenitor game *Pong*™ was already approaching 40 years of age [14–16]. Furthermore, the pervasiveness of video games for young people today in the United States can be quantified in part by noting that 97 percent of them report playing some form of video game, whether on gaming consoles, on personal computers, or, increasingly, on mobile devices [17]. The familiarity of the video game medium therefore offers the potential to tap immediately and intuitively into the everyday experience of the target audience of young sailors. The intuitive interfaces and instinctive gameplay developed for *SGD* allow players to focus on learning ship defense rather than on the mechanics of the game itself.

Technology

In tandem with the expansion of the influence of video games, the exponential growth in the computing capability that fuels the industry offers opportunities for developing instructional methods different from those found in more traditional teaching [18]. With educational video games, teachers can take advantage of immersive and engaging on-demand lessons, networked team training, and immediate examination with feedback [19]. In addition, the massive data collection and new analysis techniques supported by a modern video game permit *SGD* developers to explore new avenues for improved and adaptive teaching, training, and testing [20].

Natural Game

While the stakes are extremely high and very real, the defense of a ship against an attack of ASCMs aligns itself very well with the pure definition of a game: two independent sides (the defense and offense) with definite objectives (minimal/maximal damage), operating under certain rules (the capabilities of defensive/offensive systems) [21]. Learning to play the game translates to the core goal of *SGD*: teaching sailors how better to defend their ships in the real world. Additionally, the tense real-time scenarios faced in defense against ASCMs naturally add an element of excitement and entertainment to the game, increasing player engagement and educational opportunities.

The Reality for the Simulation

The game of chess can be intricately complex even though the movements of individual pieces are straightforward to define. In much the same way, the complexity in mounting a defense against ASCMs is derived from the much simpler definition of the offensive and defensive systems that may be employed. Understanding the capabilities provided by these "pieces" is therefore necessary for understanding the nature and magnitude of the complexity found in the overall "game" of ship defense.

The Offense

With significant roots in the technology developed near the end of World War II, the first ASCM was introduced on the world stage in the late 1950s [22]. Since that time, the diversity of ASCM types and their associated array of capabilities have grown steadily, with a world arsenal of more than 75,000 and the number of distinct types in excess of 100 varieties [23].

Though the diversity of ASCM systems is dauntingly large, the number of characteristics needed to define a given system at a high level is comparatively compact. Namely, once the flight profile (how it moves) and terminal seeker (how it sees and thinks) are defined, the system can be modeled sufficiently for the training goals in *SGD*.

Cruise missiles are kinematically diverse, with speeds ranging from subsonic to highly supersonic and altitudes from very high down to low-profile sea-skimming approaches [24]. Additionally, some systems incorporate high-g maneuvers in an attempt to evade missile interceptors fired by the defense [25]. The need for quick decision making by the defense can be brought into focus by considering fast, low-flying threats, for which the time from first appearance above the horizon of the ship until impact can be less than one minute.

Because a ship is a moving target, all cruise missiles have some sort of terminal seeker designed to guide the missile to impact its target. Many ASCMs have radars mounted in their noses for this purpose, but passive sensors (homing in on emissions from the ship) or infrared sensors are also possibilities [26]. While the seeker enables the missile to select among targets and attempt to filter out decoys, it also provides an avenue for the defense to counterattack via electronic warfare techniques [27].

The Defense

Since the advent of the ASCM threat, the Navy has continually developed and deployed a wide range of ASCM countermeasure systems. Though the diversity of systems is large, all of them can be sorted into one of two classes: hard-kill (physically destroying or disabling the threat) or soft-kill (confusing or blinding the ASCM seeker) [28].

The primary hard-kill systems aboard ships are defensive missiles called interceptors, designed to physically destroy the attacking ASCM before it can hit the ship [29]. Much like the ASCM threat, these defensive systems are defined by how far and fast they fly, as well as by the type and capability of their own terminal seekers. Additionally, close-in weapon systems utilizing a high-rate-of-fire gun are also a form of hard kill [30]. Each hard-kill system's strengths and weaknesses determine the likelihood of its effectiveness against a given threat.

In contrast to the dramatic operation of hard-kill systems, the soft-kill systems on the ship employ more subtle means to defeat incoming threats. Onboard and off-board jammers interfere with the operation of the ASCM seeker to blind or confuse its targeting, attempting to render the threat unable to guide to the target ship [31]. Additionally, soft-kill decoy countermeasures can be deployed to act as a more enticing target, causing the threat not to target the actual ship [31]. The performance of soft-kill countermeasures depends heavily on when and where they are deployed and on the capability of the seeker installed on the attacking missile.

The counter-ASCM systems, both hard and soft kill, are supported at some level by the radars on board the ship and off board (e.g., on aircraft), as well as by electronic support measure systems listening for threat seeker emissions [32].

The Game

At the most basic level, the goal for the offense is to inflict as much damage as possible (potentially on high-value ships) with its resources, while the defense seeks to mitigate the damage and conserve its own countermeasures, saving them for potential subsequent attacks.

In the conflict, the offense has some significant advantages, including deciding when the attack will occur, which types of ASCMs will be used, how many will be deployed, how they will be spaced geometrically and in time, and which ships will be targeted. The offense is challenged by two conditions: the target is moving, and the ships in the strike group can operate as a team.

The defense's advantage is that it decides which ships are in the strike group, how they are positioned, and how they are equipped. Challenges for the defense include the uncertain identification of the attacking threats and imperfect knowledge of how many threats will attack at the current time and how many may attack later.

An effective defense requires judicious employment of countermeasure systems, with correct deployment timing and doctrine, in the face of limited information on a very limited timeline. The complexity of this challenge has spurred the continued development of the *SGD* game to help sailors become familiar with the critical decisions they may face and their options.

To maximize the clarity and effectiveness of instruction, a serious game must represent the salient features of the simulated scenario while minimizing extraneous information [33]. *SGD* was designed to provide minimally detailed representations of real offensive and defensive systems while essentially retaining the full complexity of the systems' combined interactions that would be faced by a sailor mounting a defense against the broad array of potential ASCM threats.

Because the task of defending a ship against ASCMs can be seen to fit perfectly in the paradigm of a game, *SGD* was seen as a logical, relevant training tool for the modern sailor.

Genesis of Strike Group Defender

The Navy is pursuing the enhancement of capabilities across a wide variety of new and ongoing hard- and soft-kill efforts. From large programs of record for new radars and electronic warfare systems, to Future Naval Capability efforts, to speed-to-fleet reactions to urgent

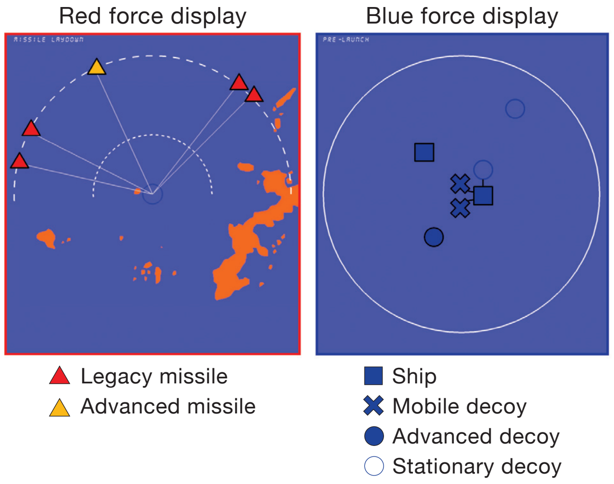


FIGURE 2. The Red/Blue game featured separate displays for the offense (left) and the defense (right).

needs, the breadth of new development is considerable [34–36]. In addition, concurrent with the creation of new systems, new tactics and deployment algorithms are being pursued. The result is that the capabilities and complexities encountered by today's sailors are steadily increasing, and so is the need for education that addresses these technological advances. Lincoln Laboratory's involvement in many of the Navy's development efforts has afforded the Laboratory insight into both the capability and training needs of naval personnel.

The first steps toward the development of an educational video game to address the evolving needs of the Navy were taken during the 2013 edition of the Lincoln Laboratory annual Red/Blue game held by the Air, Missile, and Maritime Defense Technology Division (Figure 2). In contrast to the later expanded *SGD*, the first iteration was focused on conveying several key concepts that illustrate the complexities of defense against ASCMs. This version was constructed as an intense real-time, simultaneous, two-player game, with one player acting as the offense and the other as the defense. The game was publicly introduced during an eight-team tournament conducted in conjunction with the 2013 Air and Missile Defense Technology (AMDT) Workshop at Lincoln Laboratory.

Though the game was comparatively limited in scope at this stage, constructed by engineers (not game developers) and played by engineers (not Navy ensigns), several notable findings emerged from the tournament. First, it was clear that teams that practiced more ahead of time (i.e., trained together) performed markedly better than the less practiced teams. Additionally, teams that identified a limited selection of preferred strategies on both offense and defense were better able to respond quickly to a wide range of their opponent's strategies. And finally, teams that had clear roles identified for each member were able to perform more measured responses, even against unexpected opponent behavior, on a short timeline. Though the observations from the Red/Blue tournament were qualitative, they provided insight into the potential for using a video game construct for experimentation and training.

Based on the initial demonstration of *SGD* during the Red/Blue game at the AMDT Workshop, the Office of Naval Research, PMR-51 Branch, expressed a desire to greatly expand the game into an immersive training and demonstration tool for the Navy. Partnering with professional video game developer Pipeworks and game consulting firm Metateq, Lincoln Laboratory rapidly transformed the Red/Blue game into the first version of *SGD*, which personnel at the Naval Postgraduate School in Monterey, California, beta tested.

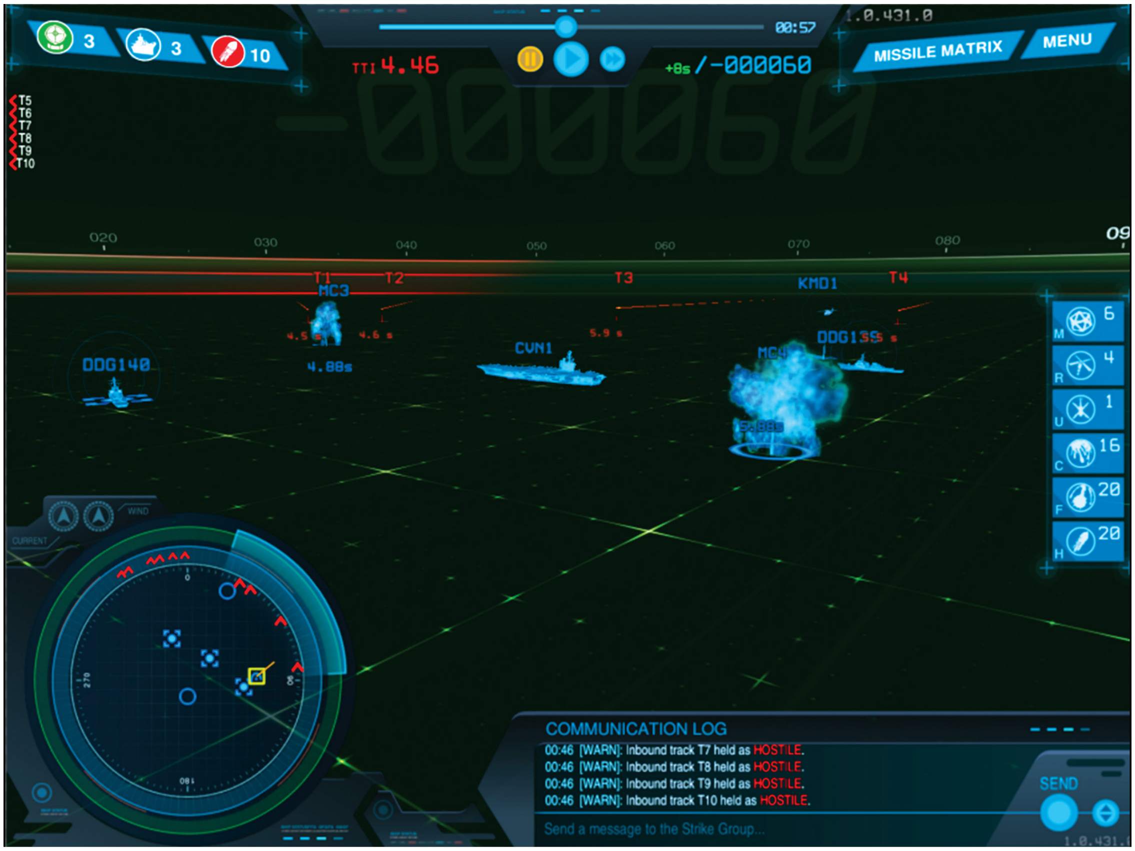
Strike Group Defender Functionality

Over a few months, Pipeworks incorporated an array of professional-grade enhancements to expand the concept of the original Red/Blue engineering demonstrator into the first polished iteration of the *SGD* video game, which offered significantly increased training potential. Immediately striking were the improved visuals and stirring soundtrack designed to create an immersive atmosphere and draw the player into the game. The action takes place in a third-person, three-dimensional arena, with the defended ships (blue) in the center and the cruise missile threats (red) flying in from the horizon. Supplementary displays and interfaces are arrayed around the large central display, providing users with easy access to all information needed to play the game (Figure 3).

The player is given complete control over the defense, deciding strategy, deploying countermeasures, and even changing ship speeds and headings. This idealization, along with the representation of the interfaces, is intended to teach core concepts rather than simulating a particular display, piece of hardware, or role for an individual sailor. The supposition underlying the game's design is that sailors more well-versed in the global operations of ship defense will better be able to fulfill their particular roles as part of the crew.

Similar to the idealization of the displays, the missiles and countermeasures in *SGD* are abstract representations intended to convey core concepts rather than represent real systems (Figure 4). On the red side, the types of ASCMs vary primarily with how the missile finds or selects its target. The Moth Missile, for example, uses an infrared seeker to measure heat from ships. On the blue side, the systems are broadly representative of classes of countermeasures. For example, the Hard-Kill system represents the full variety of hard-kill options on a ship.

However, with the understanding that more realism could be desirable for some instructional considerations, the game was designed in such a way that converting to more realistic (and therefore also classified) representations amounts to a straightforward change to the input file defining the system.



```
00:46 [WARN]: Inbound track T7 held as HOSTILE.
00:46 [WARN]: Inbound track T8 held as HOSTILE.
00:46 [WARN]: Inbound track T9 held as HOSTILE.
00:46 [WARN]: Inbound track T10 held as HOSTILE.
```

Send a message to the Strike Group...

FIGURE 3. The display for *Strike Group Defender* gameplay presents blue ships and red threats (center), an overhead view (lower left), a message panel (lower right), a countermeasure inventory (right), and menus and scoreboard (top).

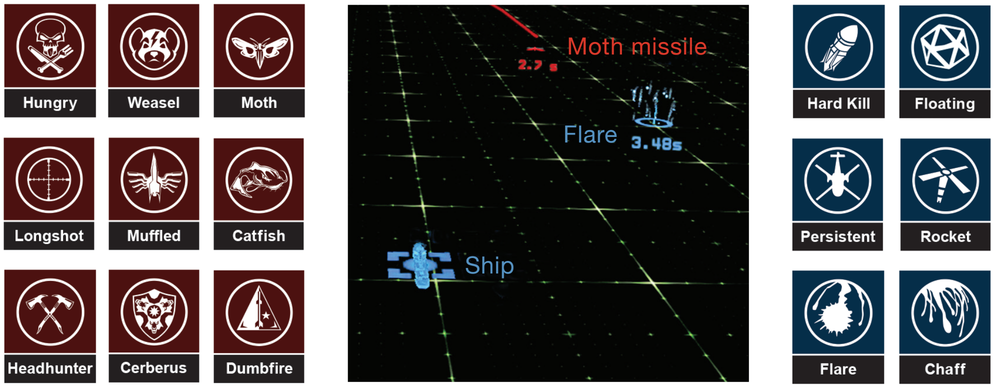
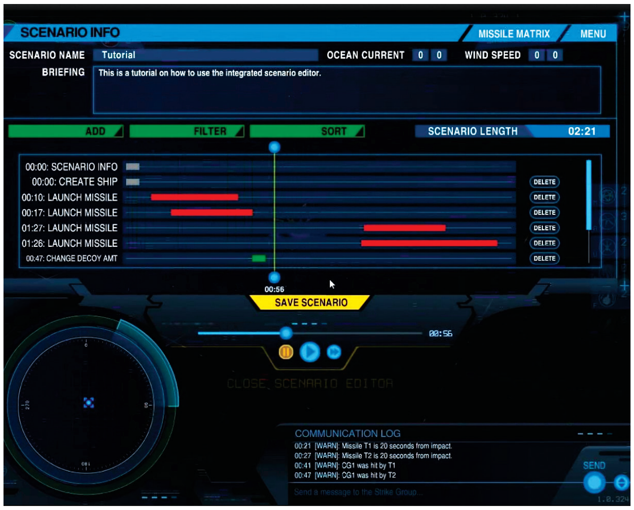


FIGURE 4. *Strike Group Defender* features a variety of abstracted threat missile types (left) and defensive countermeasures (right), each with its own characteristics and capabilities. For example, the Moth Missile can be distracted from the defended ships with a flare (center).

SGD is packaged with a range of built-in scenarios, from one-threat versus one-ship tutorials up to a full strike group versus an attack of 20 missiles or more. The game is not limited to these scenarios, however, as *SGD* also includes a built-in scenario editor that permits instructors and students alike to create their own situations (Figure 5). Ships, countermeasure loadouts (number of countermeasures carried on board), threat type, bearing, and timing are all adjustable, allowing players to explore actions from the point of view of both sides of the conflict. Additionally, threat timing and bearing can be varied, even randomly, adding challenge and discouraging rote memorization of responses.

Scenarios are played in real time, typically last a few minutes, and are playable under a variety of game modes that provide different instructional opportunities for the user:

- Tutorial. Straightforward scenarios with a single type of incoming threat coupled with a virtual instructor teach players where and when to deploy the correct countermeasures.
- Single-player defense. Controlling a single ship or group of ships, users defend against a computer-controlled ASCM attack in a variety of scenarios across a range of difficulty levels based on the number of incoming threats and the availability of countermeasure resources.
- Multiplayer defense. Through in-game text messaging or over a voice network, multiple players collaborate in real time to defend surface ships against a computer-controlled ASCM attack.
- Multiplayer offense versus defense. One player controls the adversary ASCMs (i.e., offense) while all other players collaborate as the defense. This setup enables players to gain insight into potential adversary strategies and the tactics to counter them.



SCENARIO NAME Tutorial OCEAN CURRENT 0 0 WIND SPEED 0 0

BRIEFING This is a tutorial on how to use the integrated scenario editor.

SCENARIO LENGTH 02:21

00:00: SCENARIO INFO
00:00: CREATE SHIP
00:10: LAUNCH MISSILE
00:17: LAUNCH MISSILE
01:27: LAUNCH MISSILE
01:26: LAUNCH MISSILE
00:47: CHANGE DECOY AMT

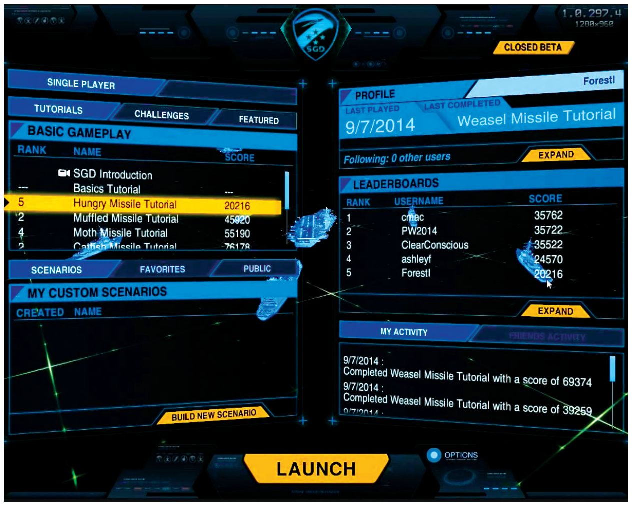
00:56

COMMUNICATION LOG

```
00:21 [WARN]: Missile T1 is 20 seconds from impact.
00:27 [WARN]: Missile T2 is 20 seconds from impact.
00:41 [WARN]: CG1 was hit by T1
00:47 [WARN]: CG1 was hit by T2
```

Send a message to the Strike Group...

FIGURE 5. The editor panel allows the player to construct a scenario, selecting which type of threats to confront, how many, and where and when they are deployed against the elected defended ships.



BASIC GAMEPLAY

RANK	NAME	SCORE
	SGD Introduction	
---	Basics Tutorial	---
5	Hungry Missile Tutorial	20216
2	Muffled Missile Tutorial	45920
4	Moth Missile Tutorial	55190
2	Catfish Missile Tutorial	76178

MY CUSTOM SCENARIOS

CREATED	NAME

Forestl

LAST PLAYED 9/7/2014 LAST COMPLETED Weasel Missile Tutorial

Following: 0 other users

LEADERBOARDS

RANK	USERNAME	SCORE
1	cmac	35762
2	PW2014	35722
3	ClearConscious	35522
4	ashleyf	24570
5	Forestl	20216

9/7/2014 :
Completed Weasel Missile Tutorial with a score of 69374
9/7/2014 :
Completed Weasel Missile Tutorial with a score of 39259
9/7/2014 :

FIGURE 6. The intuitive interface offers players easy access to scenarios, leaderboards, and social media.

In addition to the central gameplay functionality, *SGD* also incorporates many social features designed to increase player interaction, encourage competition, and ultimately improve learning (Figure 6). Each scenario has its own leaderboard on which top scores are continuously updated for all players to see. The innate desire to be atop the leaderboard is a powerful motivating force for individual improvement [37]. Similarly, the ability to create and share new scenarios with which to challenge other players is also intended to foster creativity and the desire to improve. Finally, the message board facilitates communication among the players, allowing them to ask questions of their peers and instructors and to share insights gained.

Though *SGD*'s capabilities are extensive, the game was designed from the beginning to require minimal system requirements to work properly. Running in any web browser with very low bandwidth requirements, the game retains full functionality whether played on a desktop computer in the classroom, on a laptop at home, or over secure networks on ships deployed at sea.

SGD was introduced to the wider community at the 2014 Air and Missile Defense Technology Workshop at Lincoln Laboratory. Over the three days of the workshop, 67 participants logged 332 games. The positive feedback from the community qualitatively validated many of the underlying motivations that had influenced the development of *SGD* and reinforced the Navy's desire for the research team to pursue further enhancements.

Strike Group Defender Roles

The diverse capabilities designed into *SGD* enable access to multiple training avenues with the ultimate goal of equipping sailors with the skills needed to defend their ships within complex threat scenarios. These educational opportunities can be broken down into three distinct categories: teaching, exploration, and evaluation.

Teaching

The capabilities of *SGD* enable rapid instruction in ASCM defense, from threat characteristics, to countermeasure capabilities, to implementation of correct tactics. Using the built-in capabilities, instructors can construct lesson plans to relate core concepts in the classroom setting, or students can experiment on their own.

One of the notable benefits of *SGD* is building sailors' trust in new capabilities. In response to the continually evolving ASCM threat, the Navy is rapidly introducing new countermeasure systems to the fleet. In particular, the new soft-kill systems, composed of a variety of onboard and off-board jammers, may seem arcane and untrustworthy if one does not understand how they actually do the job of defeating ASCMs. Because the new systems are unfamiliar to sailors, they may have a tendency to downplay these systems in favor of older, more familiar ones. By observing in *SGD* how new systems operate, sailors can learn how the systems work and therefore may choose to employ them appropriately in the field.

Exploration

In contrast to the cost of making a mistake in countering a real ASCM, the penalty for performing poorly in *SGD* is only a lower game score and the immediate opportunity to try to improve. This lack of consequences encourages players to experiment and to try any "what if?" scenarios desired. In this way, a deeper understanding of core concepts can be attained, and new methodologies may even be discovered [38]. Because the feedback is immediate, the trainee can try a wide variety of approaches in a short amount of time.

The game also permits outside input, which could, for example, come from another computer executing a new algorithm designed to help sailors do the job of ship defense. Thus, *SGD* can serve as a proving ground for new technologies with which sailors can interact to improve ship defense.

Evaluation

The construct of a video game, in which everything can be quantified, can provide educators with many opportunities for evaluating students' success at the tasks of the game. The *SGD* environment records a large amount of information, ranging from the number of missiles correctly mitigated, to the number of resources expended, to reaction time, to deviation from desired tactics. The availability of these data affords instructors wide latitude in evaluating the performance of *SGD* users.

Emergence of Machine Learning

As the ASCM threat has grown in numbers and complexity, so too must the capabilities of naval training grow. The fusion of a video game interface, massive data collection, and modern machine learning techniques presents a potentially powerful and nontraditional mode for enriching training for the sailors of today and the future.

The behind-the-scenes data collection built into *SGD* is no less important than the eye-catching graphics and intuitive interfaces of the game. Every action of every user in every game is seamlessly recorded into a massive data archive that allows every game to be replayed and studied by any player. This replay functionality has the benefit of allowing trainees to learn from their own successes and mistakes, and from those of other players. Beyond that, these collected data enable the game to "learn about" its players and adapt itself to their needs. Through this application of cutting-edge machine learning techniques to the *SGD* data, the instructional capability of the game is maximized, and each user is ensured an experience tailored to his or her particular learning style.

Tournament Data Collection

To demonstrate the utility of applying machine learning techniques in *SGD*, a large dataset for experimentation was needed. To fill this need, a Laboratory-wide *SGD* tournament, designated March Madness, was conducted. Beyond the intrinsic draw of competition, the Lincoln Laboratory Director's Office further encouraged participation by offering a cash prize to the champion.

Because *SGD* requires only modest computing power and functions on most any platform, the tournament could be played on demand over the local network on regular desktop computers. In the initial competition round, players attempted a variety of challenging scenarios. The top 16

What is Machine Learning?

Machine learning is a subfield of artificial intelligence in which researchers develop computational methods (i.e., algorithms) that give computers the ability to autonomously learn a model to explain data. Generally, machine learning is categorized into two branches: supervised and unsupervised.

In supervised learning, the goal is to predict an outcome from a previous example. For example, suppose a meteorologist who wants to predict whether it will rain tomorrow has data from over the previous 50 years that tells, for each day, the temperature, humidity, barometric pressure, and wind speed. These data are known as the features and are represented as a numeric vector, denoted $\vec{x}$, which describes each day. For each day, the meteorologist knows whether it rained the next day. This datum is known as the label, denoted y, for the corresponding features. The goal is then to learn a mapping f: $\vec{x} \rightarrow y$ to predict whether on any given day, described by $\vec{x}$, whether it will rain, y. This example is a classification task: the prediction variable can take on one of a finite number of values. In this case, the outcome is binary—either 0 or 1 describing whether or not it will rain. Conversely, a regression task involves predicting a continuous value, such as tomorrow's high temperature. Other examples of supervised learning include predicting whether a camera's image contains a person of interest (i.e., facial recognition), translating Arabic to English, or determining the correct medical diagnosis for a sick patient. Common techniques for supervised learning include logistic regression, decision trees, support vector machines, neural networks, and *k*-nearest-neighbors.

The complement of supervised learning is unsupervised learning. The goal of unsupervised learning is to infer a function to describe one or more hidden attributes within data. Consider an example from U.S. politics, specifically, the legislative branch. Let us say we knew that congressional representatives voted yea or nay on certain bills, and we had features describing those bills. Rather than predicting whether a representative would vote yea or nay on a future bill as in supervised learning, now we want to group representatives according to similarity. The hidden attribute is party affiliation. If we give the unsupervised learning algorithm the features of bills along with how each representative voted, the algorithm will output an assignment of each representative to a group. If this algorithm was able to mimic reality, it would learn there are three groups: Republicans, Democrats, and Independents, and it would assign each representative to one of those groups. The key is that the unsupervised learning algorithm does not know beforehand the notion of a Republican, Democrat, or Independent—just that there are groupings of some kind. In addition to our political example, other unsupervised learning tasks might involve learning taxonomy for life on Earth (i.e., how to group life by species, genus, family, etc.) or learning a grouping for people according to the types of movies they watch. Common techniques for unsupervised learning include *k*-means, Gaussian mixture models, self-organizing maps (a type of neural network), and principal component analysis.

players then competed in a single-elimination tournament; ultimately, the champion of a final-four series was crowned at an event complete with an audience and announcers.

Participation in the tournament exceeded expectations, with 140 players completing nearly 3,000 games, totaling approximately 100 hours of game time. The draw was diverse, with many players having no direct experience with the Navy in general or ship defense in particular. One player, a graphic artist who made the Sweet 16 as a top player, notably commented, "I have been illustrating these concepts for years, but now I understand what they mean." Beyond the data collected, this sort of anecdotal evidence about the benefit of *SGD* helps validate the game's educational value.

From Analysis to Enhanced Education

As players learn from *SGD*, the game is also learning from them. While the immense amount of data collected by the game holds the promise of increasing the effectiveness of instruction, it also presents an immense challenge to the analyst to distill the data down into a meaningful and useful result. Machine learning techniques are ideally suited to this situation, revealing hidden correlations and providing instructive adaptability useful to both trainees and teachers.

An exploration of the utility of these techniques began with a multidimensional analysis of the data collected from the internal Lincoln Laboratory tournament [39]. Presented here is a subset of the results of that analysis, categorized into three topics: identifying player types, identifying tactics, and adaptive lesson planning. Each of these topics has immediate relevance to meeting the Navy's educational needs for addressing complex ASCM scenarios.

Identifying Player Types

Players approach video games with a range of styles and motivations, and, similarly, there is diversity in learning approaches [40, 41]. Categorizing players based on the features measured by *SGD* is a critical first step to enable instruction that adapts to the natural tendencies of each trainee. Moreover, the characteristics of high-performing game players can be reinforced, while the approaches of lower-performing players can be identified and remedied with tailored instruction.

Critical to player typing is the collection of large amounts of data, made possible through *SGD*'s harnessing the continuous monitoring and recording of player actions enabled by modern video game technology. Which countermeasures players use and when, how quickly they react to changing situations, and which tutorials they attempted and completed and in which order, all can be used as features to help define each player.

The large number of data points and the high degree of dimensionality provided by the individual features measured by *SGD* can be reduced to a manageable set of categories through the application of unsupervised learning (i.e., clustering) techniques. The characteristics that define the players, beyond just a score or a letter grade, are then brought into focus. Players are not only categorized, but the deeper explanations for their performance can begin to be explored.

As a qualitative example of clustering, consider the classification of automobiles. The diversity in make, model, and model year is quite large, analogous to the number of players of *SGD*. Similarly, the number of features used to define a car could also be large, such as cost, performance, fuel efficiency, reliability, safety, and cargo space. Clustering could be used to determine a more concise set of automobile categories (e.g., family, utility, sport, or luxury) and the associated features that define each category. The complexity of the data is thereby reduced to a more manageable and useful level of categorization.

A more quantitative two-dimensional example of data clustering is depicted in Figure 7. Each point in the dataset is represented by two features: its x and y coordinates. The simulated input dataset is color-coded in Figure 7a to show that it was generated from five overlapping sources. In Figure 7b, the color-coding has been removed, illustrating that the underlying structure is not apparent. The challenge for effective clustering is then, given Figure 7b as an input, to extract some approximation of Figure 7a as an output.

One way to infer the true clusters underlying a dataset is to apply the k-means algorithm [42]. The process supposes a number of clusters (represented by the k in its name) and then attempts to partition the data, minimizing the cumulative distance metric seen in Equation (1).

$$\text{Cumulative distance} = \sum_{j=1}^{k} \sum_{x_i \in s_j} \left\| x_i - \mu_j \right\| \qquad (1)$$

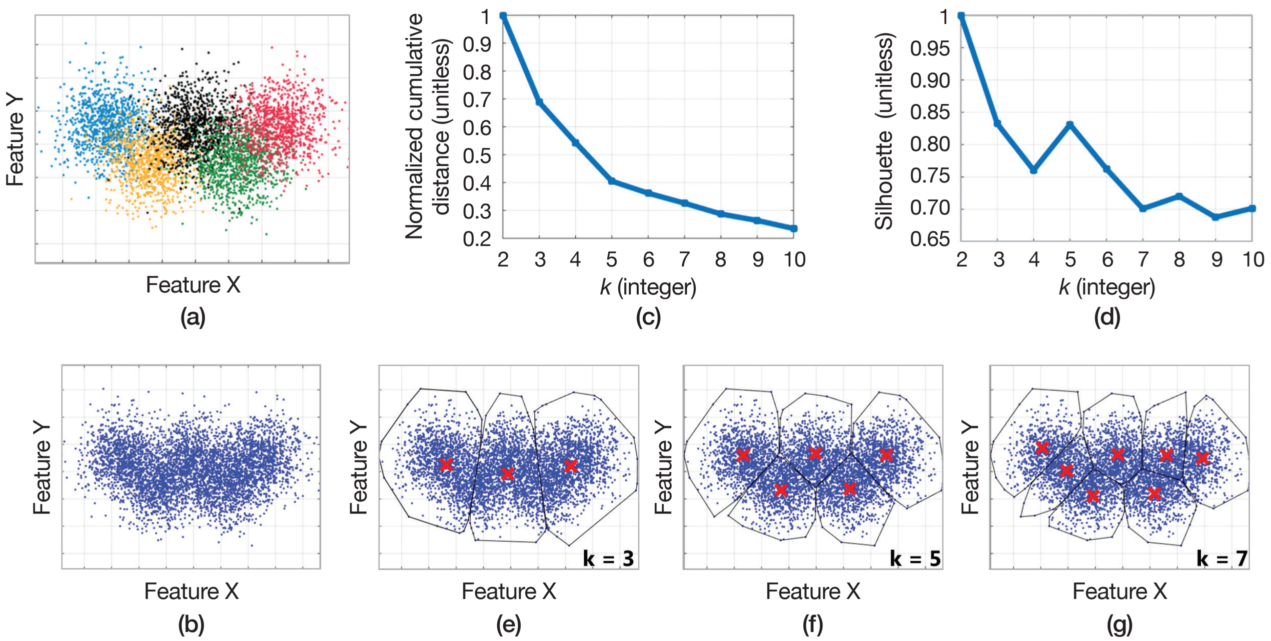


FIGURE 7. The plots show the results of *k*-means data clustering, where *k* represents the number of clusters. The dataset was generated from five overlapping sources, color-coded in (a) and with the color-coding removed in (b). In (c) and (d), the (correct) value *k* = 5 is seen to balance the goal of simultaneously achieving low cumulative distance and high silhouette (i.e., similarity of cluster members to each other). The clustering result for *k* = 5 is shown in (f), and less well-matched fits of *k* = 3 (e) and *k* = 7 (g) are shown for reference. Clusters are outlined in black, and red X's indicate the clusters' centroids.

Clustering minimizes the cumulative distance over k clusters, defined as the total distance of each data point x_i in cluster S_j from the associated centroid μ_j.

The primary output of the algorithm is the centroid of each identified data cluster, defined as the average of the features of all the points in the cluster. Each data point is closer to the centroid for its associated cluster than to any other centroid. The cumulative distance, defined as the total distance summed over all data points to their respective centers, is minimized.

The more clusters (higher k) assumed by the algorithm, the smaller the cumulative distance will be since all points will necessarily be closer to their assigned cluster centroids. Reducing the cumulative distance is a good thing up to a point, but if too many clusters are added, the whole purpose of dividing the data into more manageable partitions is lost.

Therefore, the desire to increase the number of clusters (k) is balanced against the separability power of the fit. One particular metric, known as the silhouette (Equation [2]), allows us to quantify this feature [43]. Maximizing this metric ensures that the distance from each point to the second-closest centroid is maximized.

$$\text{Silhouette} = \sum_{i=1}^{N_{points}} \frac{b_i - a_i}{\max(a_i, b_i)} \qquad (2)$$

The silhouette metric compares the cumulative distance of each data point to all other points in its associated cluster a_i to the cumulative distance to all other points in the second-closest cluster b_i.

Effective data clustering then seeks simultaneously to minimize the cumulative distance metric (Equation [1]) while maximizing the silhouette metric (Equation [2]). In Figures 7c and 7d, the value $k = 5$ can be seen to indeed best satisfy these criteria. The clustering result is shown in Figure 7f, with less well-matched fits of $k = 3$ (Figure 7e) and $k = 7$ (Figure 7g) shown for reference. The clusters are identified by the black outlines, with their respective centroids depicted with a red X.

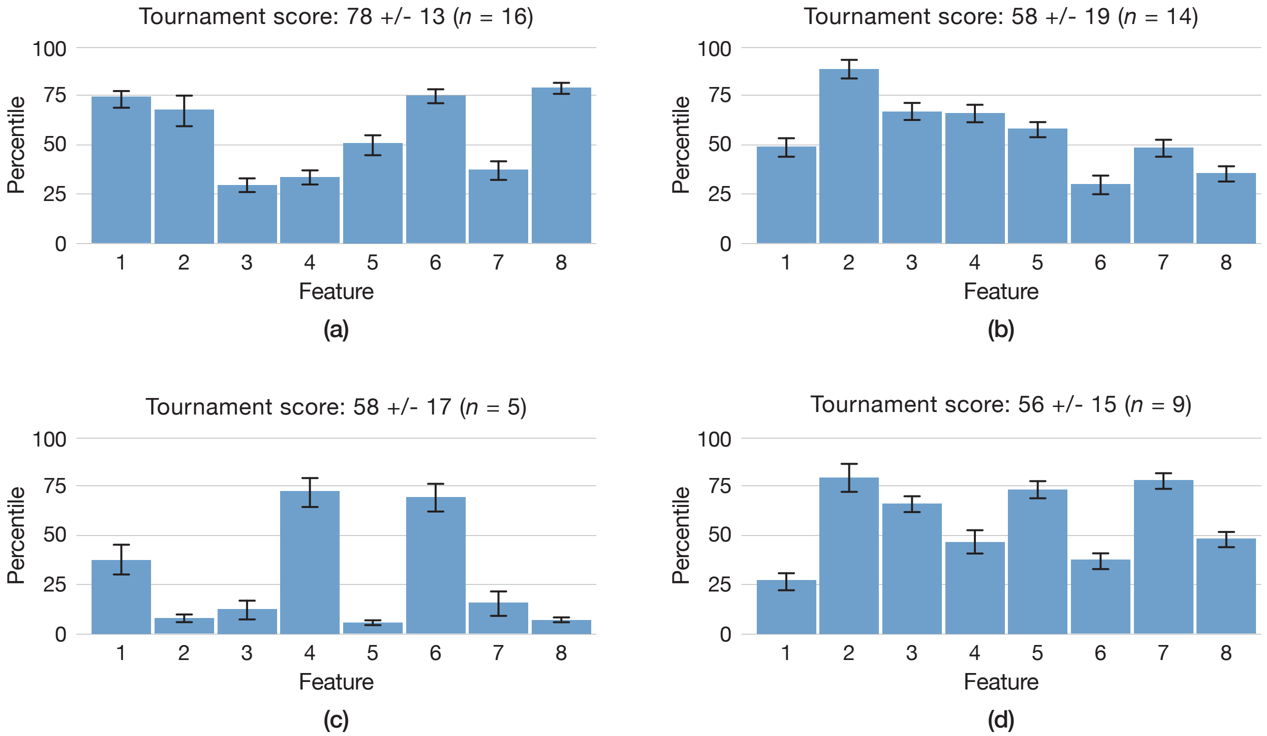
For the *SGD* tournament dataset, the following set of features was identified as most relevant for use in player typing:

1. Quit rate. Fraction of games that the player quit before the end of the scenario
2. Unique tutorials. Number of unique tutorials attempted by the player
3. Tutorial rate. Number of tutorials attempted by the player
4. Test rate. Number of times the player attempted a test level
5. Tutorials per test. Ratio of tutorial levels to test levels attempted by the player
6. Repeat rate. Number of times the player replayed a level already completed
7. Pause time. Average amount of time the player paused the game (when allowed)
8. Tutorial repeats. Mean number of times players attempted tutorial levels

Features identified with rate were normalized to the total number of games the player had played. Both the number of features (eight) and the amount of data (100 hours of gameplay) are quite large, making the exact solution of Equation (1) impractical computationally. An expectation maximization algorithm was therefore employed to allow a rapid approximation of k-means clustering to be applied to the data [44].

The result of clustering with these features on the *SGD* tournament data was the identification of four player types, shown in Figure 8. The first player type is notable for a significantly higher score in the *SGD* tournament, compared to the other three types. Paradoxically, the first player type is also distinguished by feature 1, a high rate of quitting scenarios. On the surface, this behavior would seem to be a bad quality for a player to exhibit. However, when paired with the high scenario-repeat rate also shown by this group, a play style begins to emerge: when players in this group discern that a scenario is going poorly, they quit and begin anew, immediately attempting to correct their mistake.

The other three groups performed similarly in the *SGD* tournament, though their play styles were different. The second and fourth player types both played a high rate of tutorials, differentiated mainly by the second group opting to quit scenarios while the fourth group used the pause feature more often. The third group eschewed almost all training and jumped right into the



Tournament score: 78 +/- 13 (*n* = 16)

(a)

Tournament score: 58 +/- 19 (*n* = 14)

(b)

Tournament score: 58 +/- 17 (*n* = 5)

(c)

Tournament score: 56 +/- 15 (*n* = 9)

(d)

FIGURE 8. Four player types, defined by eight unique features, were found by clustering the data from the *Strike Group Defender* tournament. The first type (a) also corresponded to the highest-scoring players. The remaining types (b, c, and d) scored similarly but had very different approaches to the game, as seen in their feature profiles.

tournament, choosing to optimize their performance only on the examination levels.

Player typing gives a window into how players approach the game and which strategies produce the more desirable outcomes. For example, if group 1 is identified as the preferred way for players to perform, lesson plans could be developed to foster the characteristics of this group in all players. Hand-in-hand with this approach, players quickly can be assigned a type as they play the game, allowing early intervention to either encourage their current approach or to correct unwanted characteristics. Through rapid player typing, the opportunities to improve the performance, and thus the training, of *SGD* users are increased. For instructors in diverse educational settings, similar player typing could inform the development of lesson plans that use video games.

Identifying Tactics

Just as clustering can be used to distill player behaviors down to a few manageable categories, it also can be applied to discover the general classes of tactics employed by players for a given scenario. The results can be used in a traditional educational sense, with instructors confirming that the trainees are indeed employing the tactics that they have been taught. Additionally, the process also allows information to flow the other way: the game can learn interesting nonstandard tactics from the players. The large number of players, combined with the freedom afforded in the gameplay of *SGD*, allows the potential for the creation of enhancements to standard tactical approaches. Thus, identifying player tactics enables improvement of both the trainees and the educational information itself.

In the application of clustering algorithms to the identification of player tactics, the features to be considered present additional complexities: time (when an action is taken) and space (the bearing of the countermeasure deployment) are integral to defining the basis feature set.

To cluster tactics, the k-medoids approach is used [45]. In contrast to k-means, where a continuum of potential centroid positions is possible for each cluster, k-medoids requires that cluster centroids be positioned precisely on an actual tactic that was employed in a particular game played. This distinction is made because it does not make sense to average individual games played to produce a "mean tactic." Put another way, deploying a countermeasure successfully to the left in one game and successfully to the right in another game does not imply that deploying it straight ahead is a viable tactic.

Like k-means, the k-medoids algorithm also seeks to minimize a cumulative distance function, as in Equation (1). However, here we are using disparate features that are difficult to compare directly. For example, deploying a rocket-type or persistent countermeasure may be seen as similar tactics, while deploying a flare would necessarily be regarded as different. To account for the variety in actions that may be taken by a player, a weighting scheme was constructed to define the comparisons among all the features making up each game [46]. With this machinery in place, the k-medoids algorithm can be applied to produce clustering results for player tactics.

To provide adequate data for clustering, participants in the *SGD* tournament were encouraged to play the Daily Performance Evaluation scenario, in which threat types were randomized for each game. The tactics extracted from the Daily Performance Evaluation data were found to cluster into four groups (not necessarily corresponding to the four player-type clusters). The prototypical tactic for each group is shown in Figure 9.

The rings around the overhead depiction of a ship represent time in the scenario, with the start time at the innermost ring and the end of the scenario occurring at the outermost ring. The colored lines indicate which countermeasure type was deployed, and on which bearing. While the tournament scores are similar, the tactics are ordered with increasingly successful performance from left to right.

The rightmost tactic came to be known as the Iron Triangle, independently identified by those that played the game. Here, the long-lived countermeasures, such as persistent or floating decoys, are deployed in a triangle around the defended ship to address a range of threats, with the player left to focus on deploying expendable countermeasures as needed to address threats not otherwise defeated. The middle two techniques are variants on this theme, with a few more countermeasures used in the second one and with the geometry a little off in the third. In contrast to the other more measured approaches, tactic number 1 is more sporadic. Recognized as an inefficient "kitchen-sink" approach, large numbers of all countermeasure types are applied continuously against the threats.

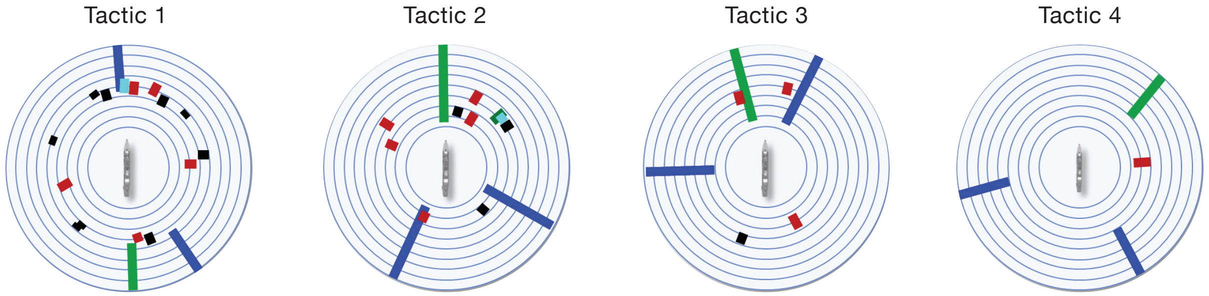


FIGURE 9. The representative player tactics derived from *k*-medoids clustering are illustrated by the four circles. Each concentric ring indicates a time in the scenario, and the color-coded lines indicate the type of countermeasure deployed. From left to right, the tactics are ordered according to increasing effectiveness.

In the context of the scenario analyzed here, the identification of tactics allows for adaptive instruction to encourage players who are already using tactic 4, to prompt players to tweak their tactics if they are using approach 2 or 3, or to teach players to totally overhaul their approach if they are using tactic 1. For other, even more complex scenarios, it is possible that tactics not previously identified as favorable could emerge, helping the game learn the preferred tactic from the players themselves.

Adaptive Lesson Plan

The continuous collection of data by *SGD* enables instructional adaptation in response to the changing needs of each player. Essentially, the game can learn how its players learn and use that information to improve its own teaching. Much as real teachers tailor their instruction to meet student needs, so too can the game adapt its interactions with the players to improve its instructional effectiveness.

As an illustration of the utility of the adaptive teaching concept, the *SGD* tournament data were analyzed to construct an adaptive lesson planner, one that could guide players through tutorials and tests on a path to maximize learning. Through contrasting the learning approaches of the lower- and higher-performing players, preferred approaches were identified. Ultimately, this approach is intended to enable the creation of an on-demand, personalized virtual instructor, one that can observe if a student is headed down the right path and give reinforcement or give correction if the student has gone astray. The potential for instruction tailored to individual students is of considerable interest to the education community [47].

While human learning is a very complex process, significant progress toward a viable virtual instructor can be made with a tractable simplified model of a person's learning [48]. To that end, a hidden Markov model (HMM) was applied to the data collected in the *SGD* tournament [49]. In this type of model, observable states, with transitions between them, are mediated by unobserved states, hence the "hidden" in the name. The model seeks to quantify transition probabilities among the states, allowing for evolution of the system to be predicted.

In the context of the model applied to *SGD*, the observable states are identified as the various tutorial and game levels available to the players. One can train an HMM on particular player types and, because the HMM is generative, create an ordered list of likely game levels. By training the model on high-performing players, game developers can create a positive lesson plan (i.e., a sequence of lessons). Similarly, by training on lower-performing players, a poor lesson plan can be produced. Players who are seen to naturally be on a positive plan can be encouraged while those on a less optimal plan can be redirected.

The HMM topology applied to the *SGD* data is depicted in Figure 10. The observable states, the tutorials, and game scenarios are depicted in the boxes at the bottom. The hidden states, which imply the players' unobservable inner machinations, are represented by the three circles at the top, designated X_1, X_2, and X_3.

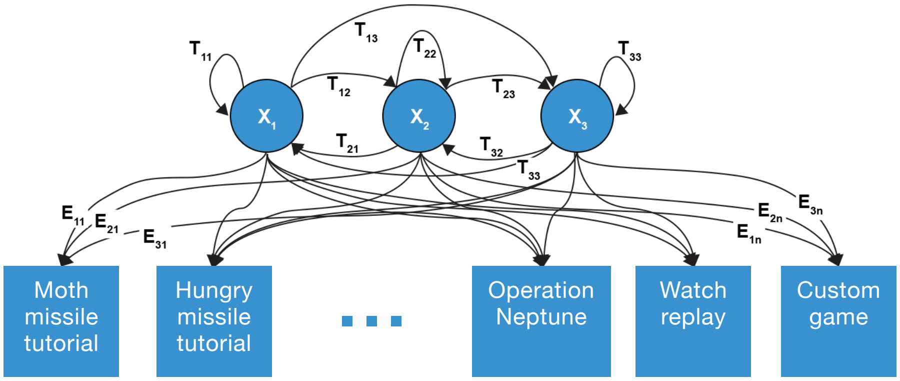


FIGURE 10. This hidden Markov model was employed to quantify the way players traverse *Strike Group Defender*, moving among the different scenarios and games represented by the lower boxes.

Table 1. Lesson Plans Depicting the Actions of Two Groups of *SGD* Players

LESSON PLAN GENERATED WITH DATA FROM BOTTOM 50% OF PLAYERS	LESSON PLAN GENERATED WITH DATA FROM TOP 50% OF PLAYERS
Basics Tutorial	Basics Tutorial
Missile Type 1 Tutorial	Challenge Mission
Basics Tutorial	Test
Missile Type 1 Tutorial	Challenge Mission
Missile Type 1 Tutorial	Challenge Mission

In between the circles are transition probabilities, represented by the T_{xy} lines. The transition probability to an observed state, typically known as an emission, is represented by an E_{xy} line. The values for all transitions and emissions are obtained by training the model on the data collected by *SGD*.

To create the adaptive lesson plan model from the *SGD* tournament data, the model was trained on data from two groups of players: the upper and lower 50 percent of performers, identified by tournament scores. The result is two complete hidden Markov models, one demonstrating how the higher-performing students navigate through the game levels and a corresponding model for the lower-performing students. In both cases, the models can then be used to recommend the next level for a student to attempt, given the level just completed. Two example lesson plans so generated from the models are shown in Table 1.

The poor lesson plan on the left side of the table shows players bouncing around between lower-level tutorials, likely making little progress. Players following the better lesson plan, on the right side of the table, appear quickly to jump into difficult challenges. It is possible that these generated lesson plans are merely indicators of player capability and may not directly stimulate player improvement. However, armed with this knowledge, the game itself can attempt to steer players onto an assumed positive path through suggestions about which levels to attempt next and evaluate player improvement along the way.

Though our initial lesson plans are derived from a simple model trained on limited data, they give an indication of the educational advantage that could be achieved with an adaptive instructor built into a game. Future enhancement may include injecting a modicum of recursion into the Markov model to better include effects of a player's history as he or she traverses the game. The true impact of this approach will be quantifiable through the collection of more data and measurement of the performance change in players provided with the adaptive tool.

Automating Players through Apprenticeship Scheduling

While the previous learning applications apply static analysis to improve a user's experience, imagine if one could dynamically adapt content in real time to a specific player's needs. Recently, Gombolay et al. have pioneered a method called apprenticeship scheduling that learns how to mimic scheduling tasks from expert scheduling demonstrations [50]. In *SGD*, Gombolay et al. showed that the tactical weapon assignments made by a player correspond to a multi-agent, multi-task, time-extended scheduling problem with complex dependencies, one of the most difficult scheduling categories. Using *SGD* tournament data, they were able to learn and mimic individual player behaviors autonomously within *SGD* via apprenticeship scheduling.

Having a learned scheduler opens the door for several real-time user interactions. For example, during an intense battle or a period of information overload, a player could be given prompts to deploy weapons in an expected way, as he or she would typically use them. Using a prompt framework, *SGD* can measure player responses to these suggestions and enable future studies of algorithm trust, acceptance, and reliance. An apprentice scheduler also enables the training of real-time autonomous agents that could either exploit the player's weaknesses or cooperate by offering prompts or filling in actions that the player might often neglect. Players could iterate with the automated adversary or a teammate to learn and improve upon weaknesses or to form a trusted, dynamic team. Having dynamic learning and feedback in *SGD* enables important studies on autonomy, human-machine interactions, teaming, and trust.

Next Steps

Machine learning techniques have a voracious appetite for data, and the studies undertaken with *SGD* are no different. As more people play the game, the dataset for analysis will grow, and the models based on it will become correspondingly better. Additionally, more data will lead to a more quantifiable assessment of the true benefits of the education tools provided by the game.

To date, the data used to explore machine learning concepts have been based primarily on the *SGD* tournament dataset. While much progress has been made, these data were collected on Lincoln Laboratory employees rather than on the true final audience, the sailors in the fleet. Expansion into this area is being facilitated by the Naval Postgraduate School, which has made *SGD* available for play by anyone in the military. The data collected from this forum can be analyzed in the same way as those from the Laboratory's tournament, and it will be illuminating to compare and contrast the extracted results.

In recent months, several improvements have been made to the *SGD* back-end to support interactions with external simulations and artificial intelligence (AI). An application programming interface (API) has been designed to accommodate external models, simulations, and decisions. Enhancements with the API include the ability to send customized prompts to a player and the ability to control the *SGD* simulation time step. Efforts are under way to reduce the simulation runtime to enable AI routines that rely on running many *SGD* instances in order to make a decision. All of these improvements can be combined with machine learning concepts to create a dynamic, adaptive learning environment not available in the Navy today.

While the back-end development of machine learning techniques and AI has been ongoing, the front-end video game has also undergone considerable development (Figure 11). The tactical focus of the first version of *SGD* has been expanded dramatically to include scenarios that require pre-attack strategizing. Full missions take place on a world map. Intelligence, surveillance, and reconnaissance (ISR) resources are built into this updated version, and new scenarios challenge players to avoid the threat of ASCMs in the first place. However, if missiles are launched in the game, the players are drawn into the original tactical-view version of *SGD*, attempting to defend their ships.

Also under development is a classified version of the game that allows for more realistic scenarios to be represented. With new scenarios and mission contexts, new weapon and sensor capabilities can be prototyped and assessed at a high level. In future versions of *SGD*, a player could configure a ship loadout or "purchase" a new weapon or AI capability and determine how well it supports the mission. Recorded player choices could also be used offline to seed an algorithm that solves for optimal loadouts and configurations. With the incorporation of these enhancements, *SGD* is envisioned as transforming from a pure focus on ASCM defense to a broader learning and technology development ecosystem that will enable the exploration of a wide variety of issues for the Navy.

Future Directions

The current research into the benefits of machine learning paired with the *SGD* platform provide a window into the training envisioned for the future. Identification of player types, for example, will help the Navy identify skilled players and also indicate ways to improve the performance of lesser-skilled players. Similarly, the identification of tactics will help identify which responses are effective, with real potential to also harness the creativity of sailors and to learn from them. The adaptive lesson plan personalizes

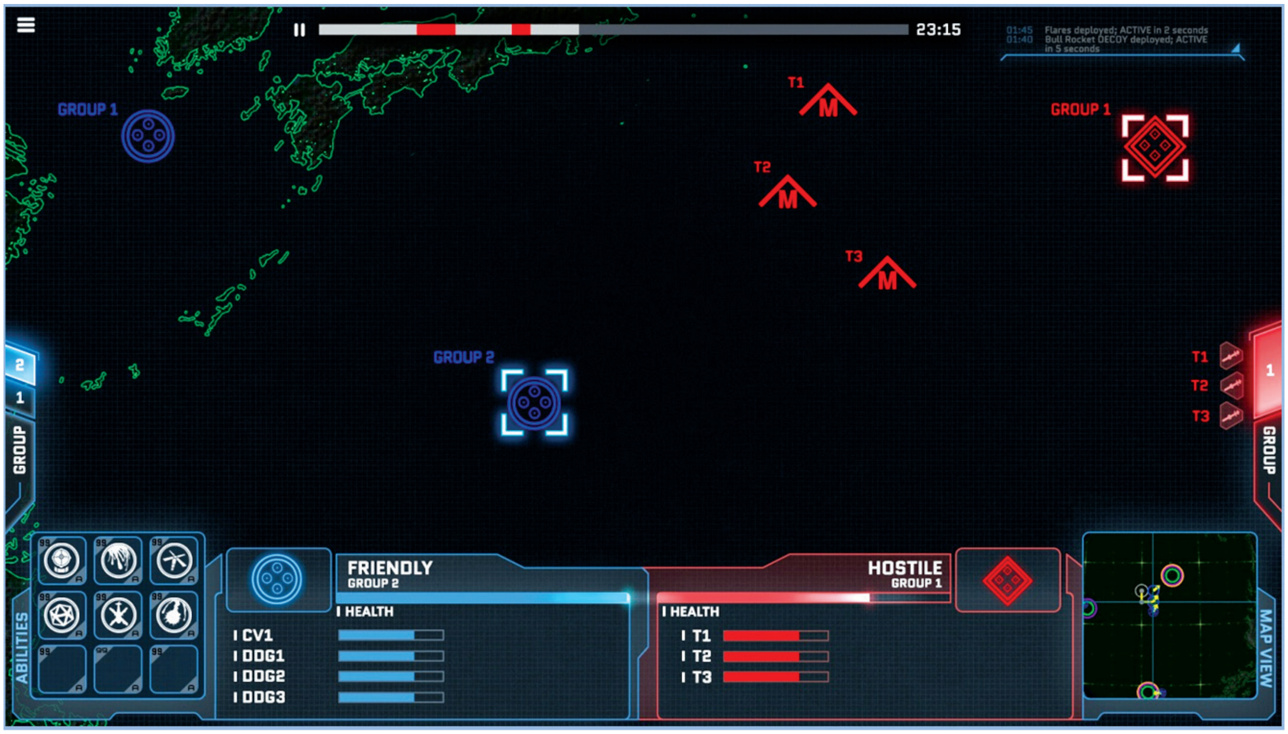

01:45 Flares deployed; ACTIVE in 2 seconds
01:40 Bull Rocket DECOY deployed; ACTIVE in 5 seconds

FIGURE 11. The next iteration of *Strike Group Defender* adds a strategic layer to the original game, with players maneuvering on a world map, attempting to avoid anti-ship cruise missile confrontations.

the learning experience for each player, offering a path to more efficient and focused instruction. The incorporation of autonomy and apprenticeship scheduling enables real-time, adaptive learning that can be tailored to players' needs. These concepts, coupled with a host of other machine learning–enabled approaches, represent a new level of training customization and engagement.

Over a short time period, the original internal Lincoln Laboratory Red/Blue game has been developed into *Strike Group Defender*, a professional video game that is coupled to back-end data storage, extended by an external API, and enhanced by AI and machine learning techniques. This combination is opening up new avenues of instruction and the ability to quantify effectiveness through the analysis of very large sets of collected data. The ultimate goal is to be able to say confidently that we have equipped the sailors in harm's way with the knowledge and skills necessary to address the threats found in challenging modern scenarios.

Awards and Recognitions

The *SGD*'s professional video game development, founded in sound technical concepts and coupled with machine learning technology, has led to recognition for the game by several government and commercial entities. In 2014, the game was recognized as the Best Government Game at the Serious Games Challenge and Showcase at the national Interservice/Industry Training, Simulation and Education Conference. The following year, the National Training and Simulation Association honored *SGD* with the team award for best training game. Finally, the MOVES (Modeling, Virtual Environments and Simulation) Institute at the Naval Postgraduate School in Monterey, California, has said in an assessment of *SGD*: "We recommend the Navy take advantage of this advancement in technology and training consistent with the recommendations being developed and put forward by the Navy Warfare Development Command (Chief of Naval Operations designated lead for Electromagnetic Maneuver Warfare)" [51].

Acknowledgments

The development of *SGD* was sponsored by the Office of Naval Research, PMR-51, with Scott Orosz guiding the work throughout, ensuring that *SGD* was developed into

a tool that would meet important needs for the Navy. Russel Phelps of Metateq was instrumental in guiding the team, enabling the engineers at Lincoln Laboratory to interface seamlessly with the developers at Pipeworks and the sponsor. The team at Pipeworks, including Lindsay Gupton, Forest Ingram, and Simon Strange, quickly and expertly changed the modest Red/Blue game into a challenging, useful, and even entertaining serious game. Perry McDowell of the Naval Postgraduate School provided useful insight, ensuring *SGD* was relevant for the real needs of Navy instructors. Without the contributions from these and others, *SGD* would not be on the frontier of education for the Navy today. ■

References

1. Congress of the United States, Congressional Budget Office, “An Analysis of the Navy’s Fiscal Year 2016 Shipbuilding Plan,” Oct. 2015, p. 3.
2. S. Saunders, “United States,” *Jane’s Fighting Ships*, Alexandria, Va.: IHS Global Limited, 7 Aug. 2015, pp. 921–960.
3. A. Feickert, “Cruise Missile Proliferation,” Congressional Research Service Report for Congress RS21252, 28 July 2005.
4. D.M. Gormley, *Missile Contagion: Cruise Missile Proliferation and the Threat to International Security*, Appendix A, pp. 177–180. Westport, Conn.: Praeger Security International, 2008.
5. M. Meyers, “Future Naval Capabilities (FNC) Program Overview,” briefing given at Naval Future Force Science and Technology Expo, Washington, D.C., 4–5 Feb. 2015.
6. Naval Education and Training Command, “Strategic Plan 2013–2023,” 2013, p.6.
7. Admiral J.M. Richardson, Chief of Naval Operations, “A Design for Maintaining Maritime Superiority,” Version 1.0, U.S. Navy release, 5 Jan. 2016, p. 7.
8. S.S. Adkins, “The 2016–2021 Global Game-Based Learning Market,” Serious Play Conference, University of North Carolina, Chapel Hill, N.C., 26 July 2016, p.7.
9. T. DiChristopher, “Digital Gaming Sales Hit Record $61 Billion in 2015,” CNBC, 26 January 2016.
10. J.W. Rice, “New Media Resistance: Barriers to Implementation of Computer Video Games in the Classroom,” *Journal of Educational Multimedia and Hypermedia*, vol. 16, no. 3, 2007, pp. 249–261.
11. J. Lussenhop, “Oregon Trail: How Three Minnesotans Forged Its Path,” City Pages, 19 January 2011.
12. A. Gershenfeld, “Mind Games,” *Scientific American*, vol. 310, 2014, pp. 54–59.
13. L.A. Annetta, “Video Games in Education: Why They Should Be Used and How They Are Being Used,” *Theory into Practice*, vol. 47, no. 3, 2008, pp. 229–239.
14. L. Clemetson, “A Nation at War: The Carrier; A Diverse Crew Reflects the Nation’s Social Changes,” *The New York Times*, 30 March 2003.
15. M. Fox, “Appendix 1: Video Game Chronology,” *The Video Games Guide*, 2nd ed. Jefferson N.C.: McFarland & Company, 2013, pp. 337–354.
16. M.J.P. Wolf, “A Brief Timeline of Video Game History,” *The Video Game Explosion: A History from PONG to Playstation and Beyond*, Westport, Conn.: Greenwood Press, p. xvii.
17. A. Lenhart, J. Kahne, E. Middaugh, A. Macgill, C. Evans, and J. Vitak, “Teens, Video Games and Civics,” Pew Research Center Report, 16 Sept. 2008, p. 2.
18. J. Simões, R.D. Redondo, and A.F. Vilas, “A Social Gamification Framework for a K–6 Learning Platform,” *Computers in Human Behavior*, vol. 29, no. 2, 2013, pp. 345–353.
19. D. Oblinger, “The Next Generation of Educational Engagement,” *Journal of Interactive Media in Education*, vol. 8, 2004, p. 14.
20. B. Marr, “Electronic Arts: Big Data in Video Gaming,” *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Chichester, U.K: Wiley & Sons, 2016, pp. 273–279.
21. C.C. Abt, “The Reunion of Action and Thought,” S*erious Games*. Lanham, Md.-London: University Press of America, 1987, pp. 3–14.
22. T. Mahnken, “Anti-Ship Cruise Missiles,” *The Cruise Missile Challenge*, Washington, D.C.: Center for Strategic and Budgetary Assessments, 2005, pp. 9–13.
23. A. Feickert, “Missile Survey: Ballistic and Cruise Missiles of Foreign Countries,” Congressional Research Service Report for Congress RL30427, 5 March 2004.
24. Naval Studies Board, “Cruise Missile Threats,” *Naval Forces’ Capability for Theater Missile Defense*, Report of the Naval Studies Board. Washington, D.C.: National Academy Press, 2001, pp. 26–30.
25. P. Zarchan, “Proportional Navigation and Weaving Targets,” *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, 1995, pp. 969–974.
26. N.F. Palumbo, “Homing Missile Guidance and Control,” *Johns Hopkins APL Technical Digest*, vol. 29, no. 1, 2010, pp. 2–8.
27. D. Adamy, “Ship-Protection Model Example,” *EW 101: A First Course in Electronic Warfare*. Norwood, Mass.: Artech House, 2001, pp. 248–250.
28. Naval Studies Board, “The Role of Hard-Kill and Soft-Kill Assets,” *Integration of Hard-Kill and Soft-Kill Systems for More Effective Fleet Air Defense*. Washington D.C.: National Academies Press, 1992, pp. 16–19.
29. W.S. Carus, “Antiship Missile Defenses,” *Cruise Missile Proliferation in the 1990s*. Westport, Conn.: Praeger, 1992, pp. 101–102.
30. M. Fuller, “Mk 15 Close-in Weapon System, Phalanx,” *Jane’s Naval Weapon Systems*, issue 55. Alexandria, Va.: IHS Global Limited, 2011, pp. 650–654.

31. D.G. Kiely, "Active Electronic Warfare—ECM," *Naval Electronic Warfare*, Brassey's Sea Power: Naval Vessels, Weapons Systems and Technology Series, Vol 5. London-Washington, D.C.: Brassey's Defence Publishers Ltd., 1988, pp. 50–77.
32. M. Streetly, "United States," *Jane's Radar and Electronic Warfare Systems*, 2011–2012," Alexandria, Va.: IHS Global Limited, 2011, pp. 150–164 & 465–469.
33. S. Thiagarajan, "The Myths and Realities of Simulations in Performance Technology," *Educational Technology*, vol. 38, no. 5, 1998, pp. 35–41.
34. T. Killion, "Future Naval Capabilities," presented at the *NDIA 15th Annual Science and Engineering Technology Conference*, College Park, Md., 9 Apr. 2014.
35. Chief of Naval Operations Directive, "Speed to Fleet Process," release from the Department of the Navy, OPNAVINST 3050.26, 9 Apr. 2015.
36. United States Navy, "U.S. Navy Program Guide 2015," Release from Department of the Navy, 2015.
37. C. Butler, "The Effect of Leaderboard Ranking on Players' Perception of Gaming Fun," *Online Communities and Social Computing*, A. Ant Ozok and P. Zaphiris, eds. Berlin-Heidelberg-London: Springer-Verlag, 2013, pp. 129–136.
38. J.J. Lee and J. Hammer, "Gamification in Education: What, How, Why Bother?" *Academic Exchange Quarterly*, vol. 15, no. 2, 2011, pp. 146–150.
39. M. Gombolay, "Machine Learning for Education: Learning to Teach," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, Nov. 2016.
40. L. Curry, "An Organization of Learning Styles Theory and Constructs," presented at the Annual Meeting of the American Educational Research Association, Montreal, Quebec, Canada, 11–15 Apr. 1983, http://eric.ed.gov/?id=ED235185.
41. M. Sailer, J. Hense, H. Mandl, and M. Klevers, "Psychological Perspectives on Motivation through Gamification," *Interaction Design and Architectures Journal*, vol. 19, 2013, pp. 28–37.
42. J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the 5th Berkeley Symposium of Mathematics, Statistics, and Probability*, vol. 1, 1967, pp. 281–297.
43. P.J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, 1987, pp. 53–65.
44. T.K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, 1996, pp. 47–60.
45. L. Kaufman and P. Rousseeuw, "Clustering by Means of Medoids," *Statistical Data Analysis Based on the L1 Norm*, 1987, pp. 405–416.
46. D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, 1993, pp. 850–863.
47. H. Gardner, "Personalized Education: A Quantum Leap in Learning Will Allow Everyone to Go to the Head of the Class," *Foreign Policy*, vol. 172, 2009, p. 86.
48. A.W. Melton, *Categories of Human Learning*. New York: Academic Press, 1964.
49. L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, 1966, pp. 1554–1563.
50. M. Gombolay, R. Jensen, J. Stigile, S-H. Son, and J. Shah, "Apprenticeship Scheduling: Learning to Schedule from Human Experts," *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 826–833.
51. P. McDowell, "Strike Group Defender Assessment," to-be-published report for the Modeling, Virtual Environments and Simulation (MOVES) Institute at the U.S. Naval Postgraduate School, Monterey, Calif.

About the Authors

G. Mark Jones is a senior technical staff member in the Advanced Concepts and Technologies Group in the Air, Missile, and Maritime Defense Technology Division at Lincoln Laboratory. While at the Laboratory, he has worked as a modeler and systems analyst on projects including electronic warfare, threat capability assessment, and ballistic missile defense. Additionally, he has contributed to a range of innovative field tests. He holds a bachelor's degree in physics from the University of Kentucky and a doctoral degree in high-energy physics from the California Institute of Technology.

Matthew C. Gombolay held a postdoctoral position in the Ballistic Missile Defense System Integration Group at Lincoln Laboratory and has now assumed a position as the Catherine M. and James E. Allchin Early-Career Assistant Professor in the School of Interactive Computing at the Georgia Institute of Technology. He researches new computational methods to facilitate human-robot interaction and optimization, focusing on harnessing the strengths of human domain experts and sophisticated computational techniques to form collaborative human-machine teams for manufacturing, healthcare, and military operations. At the Laboratory and Johns Hopkins University Applied Physics Laboratory, he developed cutting-edge planning and scheduling algorithms for ballistic and anti-ship missile defense with the U.S. Navy and Missile Defense Agency. He received a Best Technical Paper Award from the American Institute of Aeronautics and Astronautics Intelligent Systems Committee, and his work has been highlighted in media outlets such as CNN, PBS, NBC, *Harvard Business Review*, and public radio. He holds a bachelor's degree from the Department of Mechanical Engineering at Johns

Hopkins University and master's and doctoral degrees from the Department of Aeronautics and Astronautics at MIT, where he was a National Science Foundation graduate research fellow in MIT's Interactive Robotics Group.

Reed E. Jensen is a technical staff member in the Advanced Concepts and Technologies Group. He has been with Lincoln Laboratory for more than 10 years and has worked in the areas of electronic warfare systems and missile system dynamics. His recent emphasis is on autonomous control and coordination of surface and air platforms. He holds a bachelor's degree in physics from Brigham Young University and a master's degree in control theory from Northeastern University.

Steven L. Nelson is a technical staff member in the Interceptor and Sensor Technology Group. After joining Lincoln Laboratory in 2002, he has been involved in work on the Aegis Ballistic Missile Defense (BMD) System. In 2004, he became the co-lead of efforts to create an integrated discrimination architecture, which resulted in the development of the Aegis BMD Discrimination Prototyping Environment, a version of which was successfully demonstrated on the FTM-12 flight test in 2007. Most recently, he has been involved in the Aegis Sea-Based Terminal efforts, concentrating on guidance and simulation. He holds a bachelor's degree in electrical engineering and computer science from MIT and a master's degree in electrical and computer engineering from Northeastern University.

Cyber Red/Blue and Gamified Military Cyberspace Operations

Nancy L. Crabtree and Joshua A. Orr

Lincoln Laboratory researchers designed a serious game to investigate how such games could aid cyber security specialists in developing and practicing cyber defense strategies. Proof-of-concept experiments conducted with the prototype Cyber Red/Blue game yielded insights into game design and player behavior. An improved understanding of game dynamics can inform games' future development as tools for cyber security research, training, and real-world mission applications.

» **The security of the cyber domain has** grown rapidly into a major concern for the U.S. government and American society in general. The Department of Defense, National Security Agency, and Department of Homeland Security are working actively to ensure that the proper protections, situational awareness, decision support, and information-sharing mechanisms are in place to protect the U.S. critical infrastructure, including data, against major cyber attacks.

To support these government agencies in improving the nation's ability to withstand cyber attacks, MIT Lincoln Laboratory's Cyber Security and Information Sciences Division developed the Cyber Red/Blue serious gaming platform and defense-oriented game to explore the potential benefits serious gaming may provide for cyber security and to learn more about the human role in cyber defense. Cyber Red/Blue leverages the Laboratory's red team (offense) versus blue team (defense) exercise approach to explore the effectiveness of techniques and systems designed to respond to threats.

Key Aspects of the Cyber Domain

The cyber domain is an evolving human-made area of science, engineering, and practice that encompasses the hardware, software, networks, and data that drive the processing of information and the functioning of software-assisted physical devices. Because the cyber domain is human-made, many of its security challenges are different from those of the physical sciences. The rules of cyber operation can change rapidly, unlike the laws of the physical domain. Complexity in the cyber environment grows continuously, spurred by the adoption of new technologies and the ever-changing characteristics of the data these new technologies produce.

Human interaction with computers—the "human in the loop"—plays a critical role in the realization of cyber security goals, but this role is not well understood. Researchers working in cyber security need to gain a better understanding of not only where cyber security risks lie but also how humans can engage to minimize those risks.

Five key considerations for exploring human behavior in the dynamics of cyber security and operational resilience are the enterprise mission, the cyber threats to that mission, the mission-enabling infrastructure against which attacks occur, the human defenders' operational processes, and the roles that humans play in cyberspace operations. Central to these considerations is an understanding of the attack surface, which is understood as all the points at which a cyber attacker can gain access to a computer system or network.

Addressing Key Challenges

Two major areas in which serious games and gamification (the application of game-like elements to non-game activities) could enhance cyberspace operations are in the reduction of information ambiguity, often referred to as the fog of war, and the decrease in the time T to observe, orient, decide, and act (T_{OODA}) with respect to one's adversary.

Fog of war is a term used by the military to describe an operational situation in which unclear information leads to ineffective and/or inefficient decision making. Carl von Clausewitz in his 1832 book *On War* coined the term fog used in this manner and illustrated its attributes as follows [1]:

> ...[A] general in time of war is constantly bombarded by reports both true and false; by errors arising from fear or negligence or hastiness; by disobedience born of right or wrong interpretations, of ill will, of a proper or mistaken sense of duty, of laziness, or of exhaustion; and by accidents that nobody could have foreseen. In short, he is exposed to countless impressions, most of them disturbing, few of them encouraging....

John Boyd, a colonel in the U.S. Air Force, described the concept of the OODA loop in a number of briefings on military strategizing. In the most often quoted of these, delivered in 1986 [2], he said that "...in order to win, we should operate a faster tempo or rhythm than our adversaries—or, better yet, get inside [the] adversary's Observation-Orientation-Decision-Action time cycle or loop." Today, the OODA concept is widely used as a means to distill tasks into these four basic components in the study of decision making and the design of decision support systems, making it a concept central to the Cyber Red/Blue serious game.

Conflict in cyberspace, while new and technically challenging, still conforms to traditional models of conflict. As do defenders of other domains, defenders of cyberspace strive to minimize the fog of war and T_{OODA}, either deliberately or intuitively. However, the volume, velocity, and variety of operations in the cyber domain, coupled with enormous attack surfaces and the low cost to adversaries of mounting a cyber attack, make the goal of minimizing both information ambiguity and T_{OODA} very difficult with the tools available. The findings, training applications, and user interface improvements made through serious games and gamification research have the potential to greatly decrease fog of war and T_{OODA} while increasing operational efficacy in cyberspace.

Benefits of Serious Games

Cyber Red/Blue explores the idea that serious games can benefit practitioners, operational planners, and researchers of cyber security in the following ways:

- As game players, cyber security practitioners can master tools and processes through experimentation in a safe learning environment.
- Planners can think through scenarios to realize the dependencies, potential interactions, and available courses of action the game players face.
- Planners can observe gameplay and evaluate measured results of actions to gain insights that enable them to rapidly test and refine plans in a simulated environment before enacting those plans on the cyber "battlefield."
- For researchers, serious games can provide a methodology, a controlled environment, and iteration capabilities that allow them to isolate and measure aspects of cyberspace operations.

Employing game design elements into cyberspace operations' "battle management" systems may also improve human capacity to manage complex cyberspace operations. In the future, lessons learned from data collected in exercises using serious games could rapidly inform new mechanics for gamified operational counterpart systems, much like beta testing new game elements in precise market segments informs general-availability releases of personal computer games.

Cyber Red/Blue: The Platform

Cyber Red/Blue consists of a playable simulation platform and an initial prototype game. The platform offers an instrumented interface, a configurable simulated enterprise computing infrastructure, and a tool to create different game scenarios to allow human defenders to practice against automated cyber attackers in a measurable environment. Example configuration elements include network topology, the capabilities and numbers of workstations and servers in the enterprise, and courses of action that are available to players. Different game scenarios can include, for example, different kinds of cyber attacks, the incorporation of actual enterprise data, and tips and cues available to players.

The platform provides modular and extensible software models that execute predefined actions in response to player interactions with the simulated enterprise computing infrastructure environment and to player commands. The models interface with a publish/subscribe–based discrete-event simulation engine to enable a dynamic response to player actions by the simulated attacker and simulated enterprise infrastructure, and to generate recordings of the game events. Cyber Red/Blue includes emerging decision support tools that can be integrated within a unified cyber incident commander workflow.

Cyber Red/Blue: The Game

The prototype of the Cyber Red/Blue game was designed inside the platform as a defensively focused game in which the blue roles of planner and player defend against a simulated red attacker. The game addresses some of the cyber security decision support challenges of the enterprise defender in an operational environment.

During the initial experimental trial, players were presented with a fog-of-war problem: protect an enterprise environment while sifting through increasingly voluminous datasets. Players were required to interpret and respond to a large number of available logs and alerts generated by the enterprise's different computer systems in order to find the "needles in the haystack" that represented credible threats. Players applied an understanding of the situations presented to them to evaluate potential courses of action and to select the most appropriate action to initiate additional protections for the enterprise environment.

As players and planners made decisions in the game, the simulation responded, resulting in changes to the remainder of the gameplay. The combinations of player responses had impacts on the ability of the simulated operational infrastructure to support the enterprise mission. Impacts can include changes to the confidentiality, integrity, and availability of data and services, and the automated attacker's likelihood of taking control of the operational environment. For example, the players' ability to detect cyber attacks through their situational awareness capabilities directly correlated to their subsequent ability to respond to these attacks and take appropriate courses of action to prevent future attacks. These first-level impacts culminated in changes to the state of the enterprise mission.

The different aspects of gameplay were mapped to the different elements of the OODA loop process to give us a deeper understanding of the human needs in each of those areas. For example, situational awareness actions were mapped to the *observe* and *orient* elements of the OODA loop. Courses of action were mapped to the OODA loop *decide* and *act* elements. More details on these aspects are described in the later section on human-machine interface and displayed in Figure 2.

Developing Cyber Red/Blue

The development approach for Cyber Red/Blue was divided into three main phases: (1) survey existing simulation capabilities, (2) apply the survey findings to the design and construction of the platform, and (3) use the platform to create and run a game that has an instructive scenario.

Analysis of Pre-existing Capabilities

In our initial step, we surveyed six existing human interaction–based simulation approaches and graded each on four categories: focus, scope, responsiveness, and scaling cost. Note that in the survey *technical defense* refers to measuring the effectiveness of the computer defenses themselves (such as access controls, software

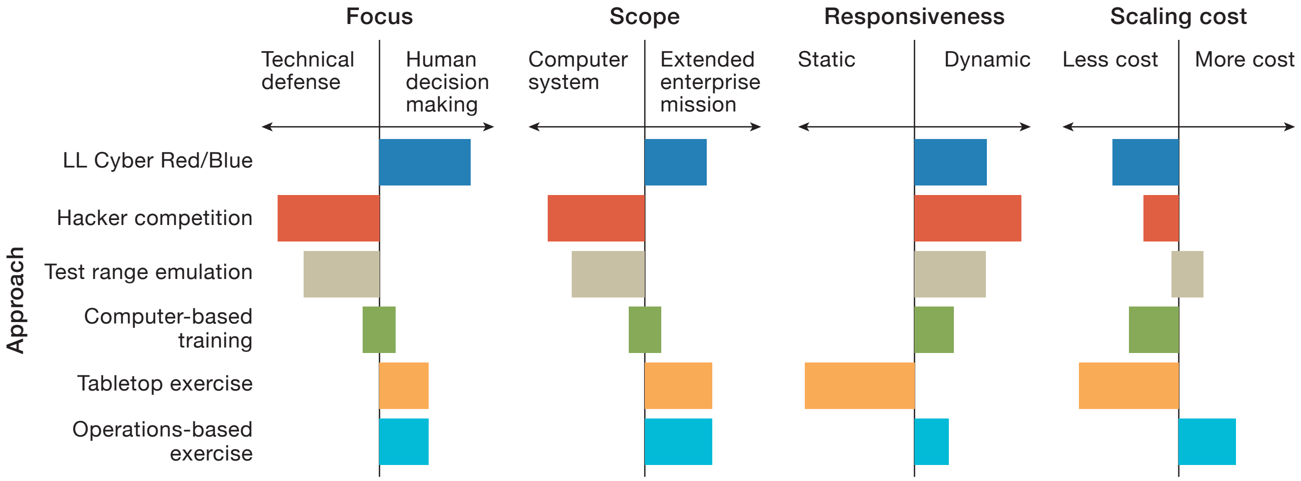


FIGURE 1. This comparison of the six simulation approaches listed on the left shows the advantage each has in the four categories listed across the top. Each category is divided into its contrasting characteristics, and the width of a colored bar indicates the relative level of advantage.

and hardware configurations, or software algorithms). In contrast, *human decision making* refers to measuring the effectiveness of the strategic and tactical approaches chosen by human decision makers (such as mission commanders responsible for making risk decisions and tasking resources at key points during the red/blue scenario). The summary findings are displayed in Figure 1.

Cyber competitions such as Capture the Flag train analysts, developers, and system administrators in a highly dynamic, emulated real-world environment through a deep emphasis on the elements of technical defense required at the computer system level. Monetary costs can be relatively low per simulation exercise instance. Computer test ranges, such as the Department of Defense National Cyber Range [3], consist of computer virtualization platforms that can be used to evaluate technical defenses of computer system interactions in a dynamic but controlled environment. Test ranges provide greater scaling capabilities than Capture the Flag but at the increased cost of a dedicated emulation environment.

At the time the survey was taken, a number of computer-based training resources were found that were oriented toward fulfilling certification and compliance requirements, and the list has expanded to include a number of online courses, such as SANS training [4] and the Department of Defense Cyber Awareness Challenge Training at Fort Gordon, Georgia [5]. Tabletop exercises emphasize the human decision-making processes of teams, but these exercises do not provide a quantitative measurement of those processes. Live operations-based exercises that make use of master scenario event lists can provide a high level of technical and decision-making realism, allowing for the wide scope of the extended enterprise mission and some dynamic outcomes, but these benefits come at significant system cost and complexity.

Developing a Needs-Based Capability

Cyber Red/Blue was designed to provide qualitative and quantitative measurement capabilities for human decision making in the context of a defensive cyberspace operation, but on a smaller scale and significantly leaner budget than the scale and budget of live operations–based exercises. The agility and low cost of the Cyber Red/Blue platform gives researchers and planners additional opportunities to experiment at more frequent intervals.

Cyber Red/Blue consists of four basic elements that are categorized as either human or automated computing components (Table 1):

1. Human-in-the-loop element. The human game players act as defenders working in a team to break the attacker's *kill chain* (i.e., a sequence of actions leading up to and including an attack). Through the game console's graphical user interface, players use simulated tools to make decisions and take actions.

Table 1. Cyber Red/Blue Simulation and Game Elements

HUMAN COMPUTING COMPONENTS		AUTOMATED COMPUTING COMPONENTS	
Human-in-the-loop element	**White cell element**	**Automated attacker element**	**Automated cyber activity element**
Attempts to break the stages of the kill chain	Develops objectives for game	Executes the stages of the kill chain 1. Undergo staging and reconnaissance 2. Gain access 3. Develop targets 4. Deploy attack 5. Verify, assess, persist in attack	Simulates enterprise environment
Decides, acts, observes, orients, as part of human-machine interface	Observes players and offers mentoring	Responds to player actions dynamically	Provides technical feedback
Utilizes technology tools to determine situational awareness, decision support, courses of action	Analyzes player activity		Creates smaller threats for game

This table summarizes key game role elements of the Cyber Red/Blue simulation. The human-in-the-loop element represents the actual game players defending the enterprise mission and its computer infrastructure. The automated attacker element is the software developed to run on the Cyber Red/Blue simulation platform that automatically executes attacks against the mission and infrastructure. The white cell element represents human analysts responsible for setting and assessing exercise outcomes. The fourth role element is automated cyber activity, which is software developed to run on the simulation platform to automatically execute the enterprise mission and its associated enterprise infrastructure background traffic.

2. Automated attacker element. The simulated attacker executes a prescribed kill chain to reach predefined objectives and is able to respond dynamically to player actions.
3. Automated cyber activity element. Configurable automated network traffic simulates the traffic of the enterprise environment that the human game players are working to protect. Through updated situational awareness indicators on the human-machine interface, this element also provides game players with feedback to inform future decisions. Additional background traffic simulates the multiple activities that can be observed in the enterprise cyber environment.
4. White cell element. Analysts responsible for setting and assessing exercise outcomes work with game planners to develop the exercise objectives. They then observe and analyze player activity to ensure the objectives are being met.

Human-Machine Interface

The human-in-the-loop element interacts with and plays the game via a role-based human-machine interface console. Figure 2 depicts the initial console layout. We did not undertake to develop a novel user interface, but rather we wanted to simply build an interface that would allow interaction with the simulated environment such that metrics could be collected. The key concept for the reader to take away from this figure is the mapping between OODA activities and potential player actions, and identification of additional tools that support evaluation during and after the game.

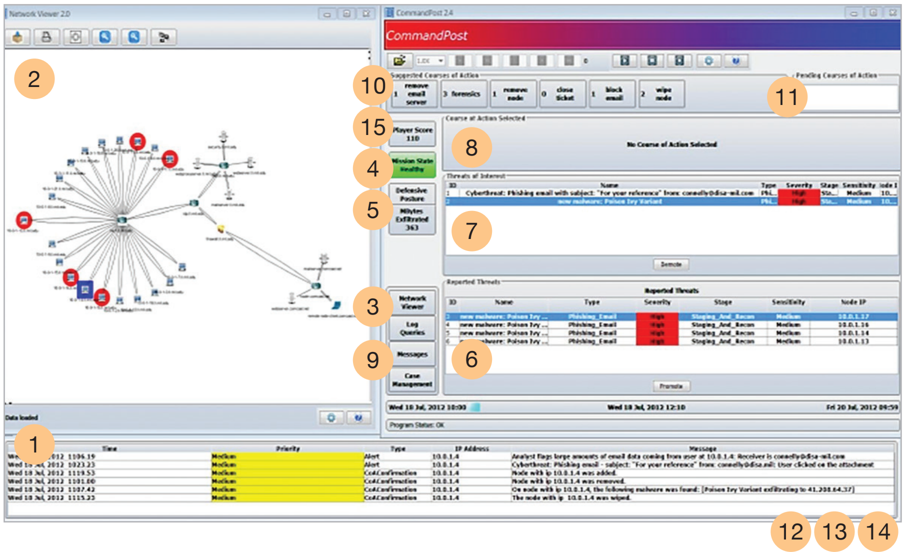
For the prototype game displayed in the figure, one example of game play function is (2) Network Display, a representation of an operational tool used for enterprise infrastructure situational awareness. The Network Display panel gives game players a diagram of the enterprise infrastructure configured for the game and updates dynamically to depict the fluctuating state of the enterprise computer infrastructure on the network. Lines depict network connections, and circles represent computers on the network. Red outlines indicate computers that have been attacked by the automated attacker, and blue outlines indicate computers that have had defensive courses of action taken on them by the game players.

Other operational capabilities displayed in the different panels include the following examples:

- Player tipping and cuing hints (e.g., Intelligence) for situational awareness provided by (1) Message Panel
- A list of identified threat types used to orient players to the mission threat environment in the context of key mission functions and guide them toward potential defense decisions, as provided by (6) Threat Context Panel
- A list of potential player courses of actions (CoAs), each with an explanation of their preconfigured risks to the mission and potential defensive contributions, as provided by (10) View CoA Costs and Benefits

Gameplay for Decision Support Challenges

One current decision challenge in the cyber domain is caused by the rapid escalation of threats. Daily, defensive operators and decision makers must parse copious amounts of uncorrelated data to find nontrivial pieces that can lead to the identification of ongoing threat activity. At the same time, information necessary to balance mission and security may be unavailable because of an incomplete understanding of the different cyber components on which the enterprise mission depends. This inefficient production and consumption of situational awareness information enables adversaries to rapidly evolve and intensify their activities without being detected when they are active, causing defenders to identify threats mostly post mortem. Once an incident is identified, decision makers must synthesize available information quickly to contain and remediate the threat and at the same time minimize mission impact. In other words, the adversary can observe, orient, decide, and act faster than today's defenders can. Attackers need only focus on their area of interest while defenders must be vigilant across the entire cyber mission



Threats of Interest

ID	Name	Type	Severity	Stage	Sensitivity	Node I
1	Cyberthreat: Phishing email with subject: "For your reference" from: connelly@disa-mil.com	Phi...	High	Sta...	Medium	10...
2	new malware: Poison Ivy Variant	Phi...	High	Sta...	Medium	10...

Reported Threats

ID	Name	Type	Severity	Stage	Sensitivity	Node IP
3	new malware: Poison Ivy ...	Phishing_Email	High	Staging_And_Recon	Medium	10.0.1.17
4	new malware: Poison Ivy ...	Phishing_Email	High	Staging_And_Recon	Medium	10.0.1.16
5	new malware: Poison Ivy ...	Phishing_Email	High	Staging_And_Recon	Medium	10.0.1.14
6	...lware: Poison Ivy ...	Phishing_Email	High	Staging_And_Recon	Medium	10.0.1.13

Time	Priority	Type	IP Address	Message
We 012 1106.19	Medium	Alert	10.0.1.4	Analyst flags large amounts of email data coming from user at 10.0.1.4: Receiver is connelly@disa-mil.com
Wed 1 Jul, 2012 1023.23	Medium	Alert	10.0.1.4	Cyberthreat: Phishing email - subject: "For your reference" from: connelly@disa.mil: User clicked on the attachment
Wed 18 Jul, 2012 1119.53	Medium	CoAConfirmation	10.0.1.4	Node with ip 10.0.1.4 was added.
Wed 18 Jul, 2012 1101.00	Medium	CoAConfirmation	10.0.1.4	Node with ip 10.0.1.4 was removed.
Wed 18 Jul, 2012 1107.42	Medium	CoAConfirmation	10.0.1.4	On node with ip 10.0.1.4, the following malware was found: [Poison Ivy Variant exfiltrating to 41.208.64.37]
Wed 18 Jul, 2012 1115.23	Medium	CoAConfirmation	10.0.1.4	The node with ip 10.0.1.4 was wiped.

Situational awareness
(Observe, orient)
1 Message panel
2 Network display
3 Log inquiries

Decision support
(Orient, decide)
4 Mission health panel
5 Threat level
6 Threat context panel
7 Threat of interest
8 Suggested course of action
9 Case management report

Courses of action (CoAs)
(Decide, act)
10 View CoA costs and benefits
11 Act by executing

White team support
12 Replay
13 Pause
14 Game/visualization control
15 Score

FIGURE 2. This figure depicts the prototype Cyber Red/Blue human-machine interface, which allows a game player to use multiple gameplay panels to execute steps from the OODA loop during the game—observe, orient, decide, and act. These OODA steps can be mapped to the different player functions, described at right of the graphic, to enable researchers to analyze player actions during and after gameplay. The mapping approach allows new gameplay panels to be swapped into different games without requiring the underlying analysis and measurement approach to change.

area to be defended and must understand how a cyber threat translates into a mission impact.

We prototyped and deployed our first game in the Cyber Red/Blue platform to test the ability of the simulation to measure human-in-the-loop capabilities while executing a game scenario. Specifically, we focused on the ability to measure game metrics related to operator environment and tools in order to understand how the environment and tools affect decision makers' fog of war and their ability to observe, orient, decide, and act. The goal of this game was to enable researchers to probe three basic questions:

1. How do various human-perceived observable artifacts (i.e., email logs, malware alerts, phishing tips, network topology, and system state) impact fog of war and T_{OODA}?
2. How do various technologies, in the form of simulated tools for situational awareness, decision support, and available courses of action, impact fog of war and T_{OODA}?
3. How do various stimuli, in the form of interactions, impact fog of war and T_{OODA}?

We tuned the Cyber Red/Blue platform to measure human-in-the-loop responses to observable artifacts by automatically tracking players' use of simulated defender tools (measured through player input to the user interface) and timing between automated stimuli and player response (measured by capturing timestamps for each event). To complement the quantitative measurements made within the system, the human analysts, i.e., the white cell element, were capable (through direct observation during the game and automatic replay of screen actions after the game) of identifying additional qualitative nuances in human perception capabilities, internal knowledge, player biases, and other psychological factors.

Initial Gameplay

For our initial game, we configured a simulated network topology consisting of server and workstation nodes on an enterprise local area network (LAN) connected to the Internet via a firewalled router. User nodes and servers executed the enterprise mission by passing email messages between themselves. The player console for the blue defender was simulated to reside on the enterprise LAN to monitor and protect the organization. The red attacker's simulated location was outside the enterprise within the Internet.

During the game, the automated attacker element simulated the red actor sending phishing emails with malicious content to blue enterprise clients. As the game progressed, some computers within the blue defender's area of responsibility were infected by the phishing emails, as represented by the red circled nodes in Figure 2. Once infected, a computer began sending out its own phishing emails and eventually started to exfiltrate mission data to the attacker.

The red attacker's goal in the game was to exfiltrate data from as many nodes as possible and to compromise the networked infrastructure by infiltrating enterprise servers from established footholds on blue enterprise nodes. Simulated attacker success meant the attacker would be capable of controlling the confidentiality, integrity, and availability of mission services. The human player's job as defender was to identify these attacks and mitigate their effects.

Throughout the game, players were presented with many observable artifacts, including a number of threats. Upon *observing* these artifacts, players could choose to "promote" threats (raise them to a higher monitoring priority level) when players determined the threats were of highest risk to the enterprise and mission. Phishing-email alerts were presented as a central threat, and players were notified of infection when nodes in the network viewer were highlighted red.

Players worked to *orient* themselves and determine the scope of the threat by performing a log query to identify other infected nodes. When players discovered 10 more infections, they decided to promote the threat. Once the threat was promoted, the player was able to view suggested courses of action and *decide* which of the actions was the most appropriate next step. To help guide gameplay, each course of action had a description of the costs and benefits to taking it.

One course of action option was to escalate the enterprise threat level, much as U.S. Armed Forces' Force Protection Conditions are elevated in response to potential threats to the nation. Other options were to block email containing specific characteristics so that the enterprise could be protected from future attacks of the same type or to remove a node from the network so that it could not communicate with other computers. Players could also simulate forensic investigations on computers to determine the underlying states of the computers and to gain an understanding of a particular threat. Additional options included wiping a node to remove malware and bringing nodes back online.

Course-of-action selection and implementation invokes some level of impact to the enterprise mission. For example, changing the enterprise threat level also changes the available courses of action. Blocking email that exhibits specific characteristics decreases the amount of email traffic exfiltrated and effectively decreases threat level, turning the mission's status panel to green to indicate mission integrity. Taking an infected email server offline stops all mission traffic and decreases mission health, indicated in red (serious mission breakdown) on the mission health panel.

In one game instance, players reviewed the infected node but did not block any additional email traffic. Because players did not choose that course of action, additional nodes became infected and the attacker exfiltrated mission data. The mission health panel changed to yellow to reflect a moderately compromised mission. After that, players had to scramble to keep up with the new level of threats. Eventually, mission health went to red because an email administrator became infected from the same phishing campaign and infected the email server.

Gameplay Findings

We played several games with separate teams of cyber researchers, security personnel, and decision commanders. We sought to create a baseline for future evaluations of decision support tools and human decision behavior, to gain feedback for improving the platform and presentation of decision support tools, and to provide insight on useful scenarios and exercise objectives.

To measure results, we first prepared the game environment by generating observable artifacts that could be measured as separate events, including operational email logs, malware alerts, and phishing tips. We configured network topology and made prototype tools available for players to monitor and control player actions during the different OODA steps. We configured the prebuilt attacks that the automated attacker would execute and the prebuilt courses of action that would be available to players at each enterprise threat level.

Before each game, we configured separate automated attack game scenarios. Each game scenario included the same enterprise and mission data, but we reconfigured the speed at which the attacks occurred to be slower at each consecutive game and increased the number of alerts that were generated in response to each attack. The consecutive game changes were necessary to allow the game players to work through the game scenario within a one-hour period.

Using instrumented results and white cell observations, we made two key findings. First, players spent most of their time on the orientation step, attempting to understand the elements of the log query tool to identify correlations between the threat context information and log query results. Second, player feedback focused on how the tools could be enhanced to improve results. Players' suggestions included adding proactive defensive capabilities to increase the security of enterprise operations before attacks occurred and enhancing the game tools to allow players to better understand attacks as they unfolded.

These findings led to several useful lessons learned:

1. "Train like you fight." We learned that for cyber serious games to be useful for practicing attacker scenarios and learning training objectives, it is important to provide the same tools and cyber environment players will face in the operational environment. In their game assessments, players focused on how the tools helped them play the game. Many recommendations from the game players related to improvements in the usability of different game console elements. This kind of feedback would be useful if we were seeking to evaluate real tools under development; however, because our tools were merely constructs intended for gameplay only, this attention to the tools diverted players' feedback from the game itself. When a game tool does not have the accuracy to emulate the real-world tool, it does not provide for the development of "muscle memory" for specific tasks, and presents the further risk that conceptual tools might inadvertently teach players the wrong lesson. These observations confirm the benefit of providing pluggable frames for inserting tools players would use in a real operational environment, especially if the game has a training objective.
2. Orientation. From our game results, it appears that without the right human decision support tools,

orientation can be the most time-consuming phase of the OODA loop in the cyber domain. Our instrumented game components allowed us to make comparisons between the times spent on the different phases of the OODA loop in order to come to that conclusion. We noted that players did not spend much time on the observe activities, such as network topology relationships and the in-depth node information available from the network viewer or from the out-of-band messages screen that collected miscellaneous enterprise information from different sources. Instead, during most of the game, players concentrated on the orient activities. Once they were oriented, players went quickly to the decide and act stages. Case management data confirmed this result. White cell members were able to observe conversations between team members to confirm that players spent the most time attempting to correlate the underlying sequence of events and did not devote much time to comparing potential course-of-action strategies for responding to the threat. Because players were viewing real operational logs but using a conceptual log correlation tool, it would be useful to perform further comparisons with real operational tools to identify the impact tools can have on condensing the orientation phase to speed up T_{OODA}.

3. Deep insight. We found that basing the game on the Lincoln Laboratory red versus blue concept could give us a multifaceted understanding of cyber decision-making processes. Our approach—which uses observable artifacts, the unified workflow, and simulated cyber models—measures multiple dimensions of player behavior simultaneously; it also provides a basis for comparing between operational tools and underlying assumptions to gain a better understanding of their impacts on defenders' success in managing challenges, such as decreasing fog of war and $T_{\mathrm{OODA.}}$

The level of abstraction was sufficient to allow players to initially track and respond to threats. While the tools were not accurate representations of specific real-world tools, they were accurate enough to reveal the lack of correlation between different cyber technologies available at the time the tests were run and the effect of this lack on the time needed to orient.

As a result, players offered a number of useful suggestions to address this lack of correlation between information elements. These suggestions included adding summarized metadata to tie system names and IP addresses back to their users, making the dependencies between the mission functions and cyber systems involved explicit, and providing transparency as to how mission health levels, costs, and benefit calculations were made. The right kind of platform instrumentation to measure human behavior on real and candidate tools, and its use to execute a game scenario and submit player feedback, could lead to a serious game (or gamification using applied serious gaming concepts) that can provide a useful format for measuring training results and evaluating the effectiveness of cyber and human tools.

The first two lessons largely validate, in a game environment, concepts that continuously plague the operational community, while the third highlights an opportunity previously unavailable and uniquely plausible in the cyber domain. How, then, might serious games begin to address these issues?

Looking Forward: Gamified Military Cyberspace Operations

Serious games like Cyber Red/Blue provide both a controlled game-like venue to answer specific experimental questions and a training sandbox. Gamification can take concepts out of the sandbox and into the operational world in hopes of achieving higher efficiency and effectiveness through "the application of game design principles in non-gaming contexts" [6]. Let's look at how gamification, informed by serious game experimentation, can begin to address these findings toward decreasing fog of war and T_{OODA}.

Train Like You Fight

Training like you fight, a concept fostered in military doctrine, leads to a soldier's development of procedural memory. For example, for pilots to learn to fly, thousands of hours of practice are needed so that they develop the reflexes that enable them to act on instinct in life-and-death combat situations. Not all of these hours can be accomplished through actual flight time because of the risks associated with flying and the resources required to send aircraft out on a training mission. Flight simulators, which are designed to emulate every detail of an aircraft and its performance, offer a way to increase training frequency and duration without the costs and risks associated with real-world flight.

Today in cyberspace operations, most hands-on technical training occurs in lab environments with real or virtual hardware and software tuned to specific training objectives without regard for the holistic operating environment (i.e., the configuration of people, processes, and technologies that make up the cyber terrain, including command and control and intelligence functions). While the use of specific programs and commands may translate from the lab into procedural memory useful in the real-world, many variables change from the classroom to the "battlefield." Introducing a common human-machine interface that employs game elements and game design to facilitate learning and efficient operation may open opportunities for the cyber equivalent to the flight simulator. Learning may be further facilitated through the use of gamified motivation techniques, such as points, badges, and leaderboards. Training in this manner may encode in procedural memory the locations and processes in software that soldiers need in order to access relevant observable artifacts and therefore decrease T_{OODA}. As many practiced players of various roles operate tools and interact more efficiently through the game-like interface, fog of war may also decrease.

Orientation

In today's cyber operations environment, orientation often requires the assimilation of information from diverse sources, distributed via multiple methods and modalities that are often nonstandard. Oftentimes, this information works its way through intermediaries that induce loss to the original information. Once real-world operators or analysts have collected and fused actionable information, it often takes hours or days to orient to the information, decide a course of action, and finally enact that course of action.

Compare the above notional T_{OODA} of real-world cyber security operations with that of the real-time strategy game *StarCraft II*®. In *StarCraft*, a casual player can sustain a productivity level of 50 complex, meaningful, and multidisciplinary actions per minute (APM) while a proficient player can sustain 300 or more APM [7]. These numbers, while unlikely in real-world operations, represent the T_{OODA} speeds humans are capable of when presented with near-lossless interfaces to accurate information, capabilities, and real-time feedback. Developing an equivalent gamified interface to real-world operations may enable players to quickly observe the artifacts presented, orient to them with computational augmentation and automation, decide courses of action based on probabilities of effectiveness, and from within the same interface take actions or issue orders and guidance for others to take action. Such a game-like interface may decrease time and signal loss from sensor to decision maker and from decision maker to actuator, thereby decreasing T_{OODA} and fog of war.

Deep Insight

While serious games tend to capture structured data regarding the impact that observable artifacts, tools, and interactions have on metrics like fog of war and T_{OODA}, these data largely go uncaptured in today's real-world operational environment. In a common gamified platform, metadata associated with each of the OODA steps can be collected and used as immediate player feedback in the form of achievement badges, experience points, and ranking on leaderboards. These metadata can also be used for analytical inquiry into the efficacy of plans developed and tactics employed in real-world operations or the exercises that precede them.

Where to Begin

In order to apply game elements and game design techniques to military cyberspace operations' mission applications, such as battle management systems, we can leverage game design approaches, such as Hunicke et al.'s mechanics, dynamics, and aesthetics framework [8].

Mechanics describes the particular components of the game, at the level of data representation and algorithms. Dynamics describes the runtime behavior of the mechanics acting on the player inputs and each other's outputs over time. Aesthetics describes the desirable emotional responses evoked in the players when they interact with the game system [8].

MECHANICS AND GAME CONTENT

All games have rules, workflows, assets, levels, roles, and a variety of other mechanisms and content that enable gameplay. To understand these mechanics for the design of a gamified battle management system for cyberspace operations, we can turn to the Doctrine for the Armed Forces of the United States, which contains thousands of pages clearly defining, among other things, the intelligence, operations, and planning methodologies employed in all domains of conflict [9]. Applying this doctrine to the cyberspace domain requires us to research the specific functions and tasks described in cyber security guides and literature. By combining the discrete tasks necessary to secure and operate networks with the military concepts necessary to conduct full-spectrum military operations, we can define the mechanics of cyberspace operations. We have already started work to describe these mechanics and expect the results to feed future prototyping efforts for a gamified battle management system.

DYNAMICS

To keep players interested, game designers often create game elements such as time pressure or tension within the storyline of the game. However, these elements already exist in real-world military conflicts. While cyberspace operations are likely to have their dynamics driven by geopolitics or current in-contact operations, we must strive to understand these and other dynamic components as we gamify the cyber operations environment. We may want to put aesthetic mechanisms in place to convey dynamics; for example, we could add countdown clocks to indicate deadlines for countermeasure deployment or audio feedback to indicate success.

AESTHETICS

To look through the eyes of the player, we must consider the aesthetics of the game and the motivations (extrinsic or intrinsic) that drive them to play the game. While in traditional military system designs aesthetics are rarely considered, they are critical in the cyberspace domain. Because of the complexity of the cyber environment, potential players will always look for ways to decrease complexity, using the path of least resistance even if doing so inadvertently increases fog of war and T_{OODA}. Designing a user interface that considers how the interface will impact the user's mental and emotional state, that is intuitive to operate, and that is even fun to use may promote the gamified system's use over more familiar systems that do not consider the game mechanics necessary to decrease fog of war and T_{OODA}. The gamified mission application should at minimum provide users a venue that makes their role easier, more effective, and more motivating than do current methods and modalities, such as email and document-based approaches.

Summary

Lincoln Laboratory's Cyber Red/Blue game environment provides a repeatable methodology for measuring human behaviors that affect cyber security outcomes. Inclusion of real operational tools in the game environment will improve training and analysis results. With actual tools and the flexible Cyber Red/Blue measurement framework, it is possible to apply additional measurement qualities of mechanics, dynamics, and aesthetics to a gamified real-world environment that simultaneously measures and trains for the future. We look forward to developing this approach further. ■

References

1. C. von Clausewitz, *On War*. Berlin: DümmlersVerlag, 1832; 1874 translation by J.J. Graham available as a Project Gutenberg ebook at www.Gutenberg.org.
2. J.R. Boyd, "Patterns of Conflict," 1986 version edited by C. Richards and C. Spinney in 2007 for the Project on Government Oversight, Defense and National Interest, www.dnipogo.org/boyd/patterns_ppt.pdf.
3. National Cyber Range, Deputy Assistant Secretary of Defense for Developmental Test & Evaluation/Director, Test Resource Management Center website, https://www.acq.osd.mil/dte-trmc/ncr.html.
4. SANS Institute website, https://www.sans.org/.
5. DoD Cyber Awareness Challenge Training, https://ia.signal.army.mil/dodiaa/.
6. K. Robson, K. Plangger, J.H. Kietzmann, I. McCarthy, and L. Pitt, "Is It All a Game? Understanding the Principles of Gamification," *Business Horizons*, vol. 58, no. 4, 2015, pp. 411–420, http://dx.doi.org/10.1016/j.bushor.2015.03.006.
7. Y. Lejacq, "How Fast Is Fast? Some Pro Gamers Make 10 Moves per Second," NBC News, 24 Oct. 2013, http://www.nbcnews.com/technology/how-fast-fast-some-pro-gamers-make-10-moves-second-8C11422946.
8. R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A Formal Approach to Game Design and Game Research," *Proceedings of the Challenges in Games AI Workshop, 19th National Conference of Artificial Intelligence*, 2004.
9. U.S. Department of Defense, *Doctrine for the Armed Forces of the United States. Washington*, D.C.: CreateSpace Independent Publishing Platform, 2013.

About the Authors

Nancy L. Crabtree has more than 13 years of experience at Lincoln Laboratory, working in the area of cyber systems engineering and architecture for both the Information Services Department and the Cyber Security and Information Sciences Division. She has applied her work in cyber security and resilience to diverse Department of Defense (DoD) domains, including installation energy, next-generation radar, space systems, command and control, acquisitions, and critical enterprise infrastructure. Prior to joining the Laboratory, she worked in software development, implementation, and quality assurance for data communications, Internet, and telephony security companies, such as GTE Internetworking, BBN, and Boston Technology. She is an early member of the regional cross-sector Advanced Cyber Security Center and a member of the Military Operations Research Society, IEEE, and several DoD information-sharing groups. She holds bachelor's and master's degrees in technical business management and organizational behavior from Thomas Edison State University and has a Certified Information Systems Security Professional certification from (ISC)^2, an international cyber security organization.

Joshua A. Orr was an application developer and cyber operations analyst in Lincoln Laboratory's Cyber Systems and Operations Group. He was assigned to the field site at Fort Meade, Maryland. In his primary role as the senior technical advisor for the U.S. Cyber Command (USCYBERCOM) Capability Development Group, he was responsible for the analysis and design of solutions to a broad range of mission-critical technology requirements for national strategic operations. He is currently the deputy director of cyber operations for the 2020 U.S. census. Prior to joining the Laboratory in 2014, he served 14 years in the U.S. Air Force in the cyber and network engineering field, supporting intelligence and cyber operations missions. He has extensive experience in cyber operations and was responsible for designing and building the Cyber Command and Control Portal for Operations while assigned to USCYBERCOM. He holds several industry certifications, including GIAC Certified Incident Handler, GIAC Penetration Tester, and Security +. He earned his bachelor's degree in business administration at Grantham University in 2011 and is currently pursuing a bachelor's degree in computer science through the University of Maryland University College.

NASPlay: A Serious Game for Air Traffic Control

Hayley J. Reynolds, Brian C. Soulliard, and Richard A. DeLaura

A serious game developed for training air traffic managers and for exploring new procedures in air traffic management enables participants to gain broad experience with traffic management decision making and the repercussions of the decisions. The game gives operators the opportunity to tackle in a day or two the decisions that they would normally encounter throughout a whole year or more.

»

In 1981, the National Airspace System (NAS) incurred a massive influx of new air traffic controllers brought on by the Reagan-era firings of more than 11,000 striking members of the Professional Air Traffic Controllers Association. As the new controllers gained experience, many advanced into positions of air traffic management, directing not only single aircraft but also large flows of many aircraft around bad weather. Because the controllers hired in 1981 are now retiring, the air traffic management domain is facing a void in experience, with most current traffic managers having five or less years of experience. If the current and new traffic managers are prepared with only "on-the-job" training, the necessary mental models of weather and traffic behavior they acquire will be based on the job experience of just five years or less. To combat this lack of experience, MIT Lincoln Laboratory researchers created an air traffic management serious game, called NASPlay, to enable a trainee to experience a year's worth of difficult days in only a day or two.

There are several reasons to train personnel through a gaming approach rather than through practice drills on a series of canned, realistic scenarios. By developing a model of the NAS (which could be scoped to various levels of complexity and fidelity depending on the purpose of the game), game designers necessarily develop hypotheses of the causes and effects of decisions within a complex environment. Because this model of the NAS can subsequently be revised on the basis of newly discovered data from the actual NAS, a more fully developed user mental model of the system can form so that cause-effect linkages between users' decisions and the resultant states of the emulated system can be clearly outlined and carried forward to the real system. Another benefit is that game players can gain decision-making experience more quickly than would be possible in the real world. A game also allows players to take risks and experiment in ways that would be unwise on the job. In addition, the gaming environment has the potential for time manipulation of the game. For example, if the user concludes that a decision was a poor one, then he or she could go "back in time" to modify or delete the action taken, resulting in a different outcome. This back-in-time capability—along with the ability to view quantitative, scored outcomes—would encourage the user to modify and optimize a strategy through iterative trial and error, and a robust scoring metric would challenge the user to replay scenarios in order to maximize scores.

Traffic Flow Management

Traffic flow management (TFM) is performed at air traffic control facilities in the NAS to assess whether the traffic demand on the system exceeds the capacity of the system. If the demand exceeds capacity, traffic management coordinators (TMCs) must decide if and how to reduce that demand. This assessment of demand and capacity imbalances takes into account NAS resources, such as runways, routes, fixes (points along routes), and sectors. Options to reduce demand include delaying departure of flights (ground delay programs or GDPs), stopping the departure of flights to a particular airport altogether (ground stops), putting flights into holding patterns in the air, or changing the route that flights have requested. Airlines may also request that flights land at an airport different from the one planned (diversions) or cancel flights that they cannot complete because of capacity and resource constraints. At the national level, mass movements of traffic demand can occur if capacity is reduced across a large segment of the United States, as often happens during large thunderstorm fronts. Options that the national TMCs have include airspace flow programs (AFPs), which reduce the traffic demand over large airspace segments by delaying the departures of any flight flying through a Flow Control Area. Some examples of Flow Control Areas are shown in Figure 1.

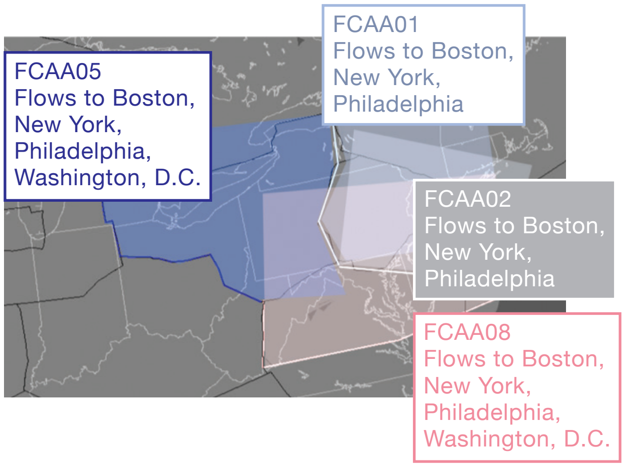

FCAA01
Flows to Boston, New York, Philadelphia
FCAA05
Flows to Boston, New York, Philadelphia, Washington, D.C.
FCAA02
Flows to Boston, New York, Philadelphia
FCAA08
Flows to Boston, New York, Philadelphia, Washington, D.C.

FIGURE 1. In the air traffic manager's computer display, the Flow Control Areas are demarked by lines "in the sky" that provide a means to control the rate of traffic traveling either north/south or east/west.

National TMCs also have the option to reroute flights, using common strategic reroutes available in the National Severe Weather Playbook [1]. Examples of different scopes of reroutes are shown in Figure 2.

To assess the demand and capacity imbalances that are present across NAS resources, TMCs have several information systems available to them. Traffic demand for a resource can be assessed through either the Traffic Situation Display or the Flight Schedule Monitor, shown in Figure 3. The Traffic Situation Display allows the TMC to visually identify the points of possible congestion, both at the current time and the time projected into the future, through either a manual time slider or an automatic movie-like projection. The Flight Schedule Monitor's set time bins, which span into the future, provide aggregated counts of flights demanding a particular resource (e.g., airport or FCA). Fair-weather capacity for resources is shown in a horizontal line to enable TMCs to easily evaluate demand and capacity imbalance.

To determine if capacity has been impacted by weather, several weather tools are available. The most critical weather tool to assess convective weather (conditions that lead to thunderstorms) out to 2 hours into the future is the Corridor Integrated Weather System (CIWS) [2], which provides convective weather information and 0- to 2-hour forecasts covering the United States and southern Canada. The Consolidated Storm Prediction for Aviation (CoSPA) is CIWS's strategic counterpart, a

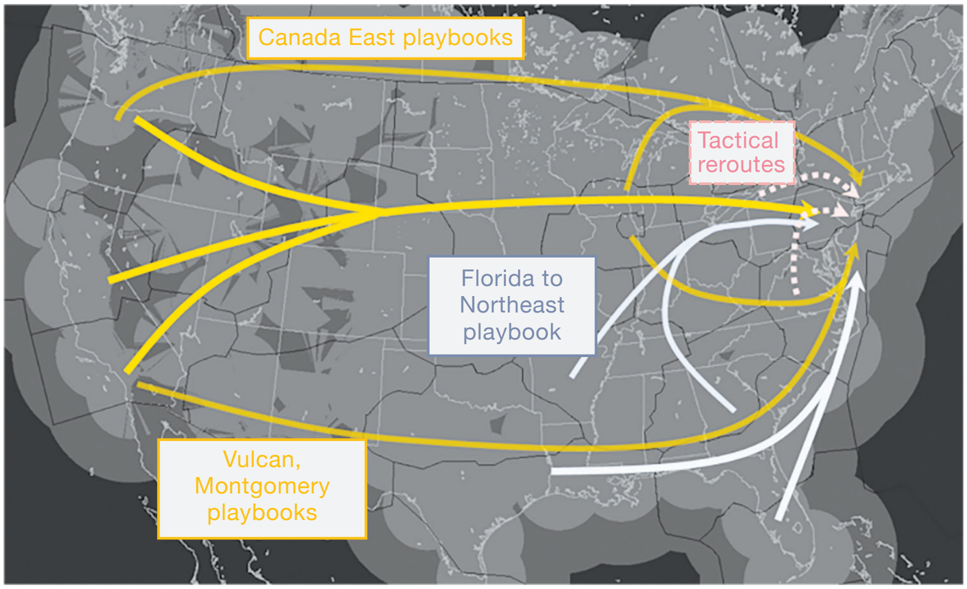
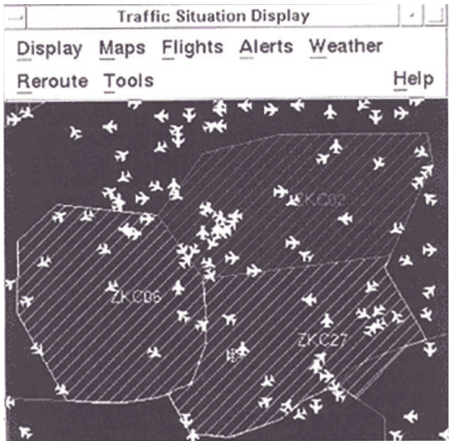
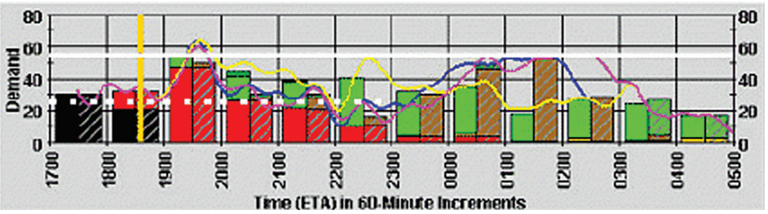


FIGURE 2. National severe weather reroute options are available to traffic management coordinators. Each colored line represents a different option in the National Severe Weather Playbook.



(a)



(b)

FIGURE 3. The Traffic Situation Display (a) provides information to traffic management coordinators about points at which airspace congestion is possible. In the figure, the polygons represent different air traffic control sectors predicted to be congested, and the aircraft icons symbolize the flights contributing to the congestion. The Flight Schedule Monitor (b) tracks demand for resources, such as airports or routes, in color-coded bins. Black represents flights that have already landed, red represents flights that are in the air currently, and green represents flights that have not yet departed.

prototype that provides deterministic weather projection 0 to 8 hours into the future [3]. The CIWS and CoSPA displays are shown in Figure 4.

The TMCs face several challenges in assessing and addressing demand and capacity imbalances. It can be difficult to evaluate the impact of adverse weather on route capacity. The varying severity of weather, pilots' unwillingness to fly through bad weather, and the inherent uncertainty in forecasts of weather and traffic demand all contribute to making the prediction of weather's impact on capacity an art. The Federal Aviation Administration (FAA) has a strong interest in keeping the NAS running at full capacity because any under-delivery of traffic costs the airlines, the FAA's "customers," money. In addition, multiple TMCs at different air traffic facilities can choose to address the demand in different, and often overlapping or conflicting, ways. Again, it is an art to determine which problems should be addressed at the national level (i.e., through AFPs and playbook reroutes) and which should be addressed tactically (i.e., through holding, ground stops, and tactical reroutes).

Applications of Gaming to Traffic Flow Management

Lincoln Laboratory's investigation into the benefits of a gaming approach to training revealed obvious applications of gaming to issues plaguing traffic flow management. Several issues with the NAS's TFM had been identified during the prototyping efforts on CIWS, the Route Availability Planning Tool (RAPT), Integrated Departure Route Planning, and CoSPA [4, 5]. Firstly, across the NAS, basic concepts of TFM are misunderstood by TMCs at the FAA facilities. To inform their decision making, TMCs are constantly striving for information that predicts evolving weather conditions as far in advance as possible; however, when more reliable information becomes available, they often fail to revisit their strategic decisions in tactical timeframes. In convective weather situations, when decision making has the most impact on traffic flow, TMCs often do not regularly reevaluate their strategic and tactical decisions to account for the fast-changing situation. Frequently, TMCs make decisions too early or too late, and they do not base decisions on the most diagnostic information available to them, preferring to rely on familiar information sources. In addition, traffic management experience is acquired slowly. Each day of convective weather provides only a single data point for the TMC to add to his or her experience. Moreover, traffic managers may not have the opportunity to learn from their decisions; for example, a TMC may make a decision at 8:00 a.m. but be off-shift before that decision shows (or does not show) results at 6:00 p.m. As researchers seek ways to improve TMC training, human-in-the-loop experiments to assess new training methods can be costly and TFM components can be difficult to replicate in the laboratory. Furthermore, because human-in-the-loop experiments are usually run in real time, they are time-constrained to address only a limited number of the wide range of scenarios that are likely to be encountered in real operations.

The gaming approach provides a means of addressing these TFM issues, particularly TMC training. Currently, training is performed in a classroom setting with TFM concepts conveyed in PowerPoint. The concepts that the lecturer is covering may or may not be directly connected with real traffic management scenarios or data and the information sources available to TMCs. A gaming environment could not only ensure that the TFM concepts are directly connected with NAS data and the information sources available to TMCs, but the TMCs could also experience dozens of situations that demonstrate the concepts. This immersive form of training could better encourage the development of strategic heuristics that, if executed correctly, could transfer positively to the real-world environment. Gaming is also a natural extension of the methods used to train air traffic controllers, who spend hours working a simulated traffic control environment.

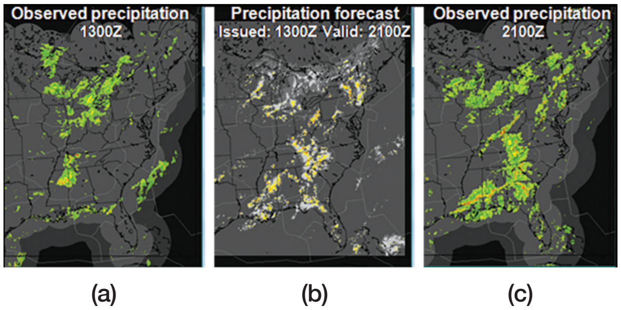


FIGURE 4. The Corridor Integrated Weather System (CIWS) shows the observed precipitation at 1300 zulu[1] (a). The CoSPA precipitation forecast issued at 1300 zulu for 2100 zulu is shown in (b). The actual precipitation at 2100 zulu is shown in (c). In the (a) and (c), green indicates light precipitation; yellow indicates medium precipitation, and orange/red is heavy precipitation. In (b), white represents light precipitation; yellow is medium precipitation, and red is heavy precipitation. While the forecast did not capture much of the light precipitation, it did capture the heavy precipitation and characterized well the type of storm (patchy, heavy in some areas, clear in others as opposed to a stationary impenetrable front). This forecast enables the air traffic manager to have sufficient information to make reroutes around the heavier precipitation.

Another TFM application of gaming is in the exploration of new procedures in consideration for their implementation into the NAS. Often, a new procedure is proposed by a TMC and then is modified and honed during actual operations, to the potential detriment of the flights exposed during the early, less effective iterations. If a game environment was available, new procedures could be explored to identify when to enact or retract a procedure, which traffic flows to target, and what amount of action to implement (e.g., what number of miles should separate aircraft) without adversely affecting any flights in the NAS until the procedure is mature. A third application is concept development for TFM products. New tools developed for TFM purposes could be tested and iterated in TFM games before being deployed into the field.

[1] Zulu is a term used to indicate that the time referenced is the Coordinated Universal Time (in essence the same as Greenwich Mean Time). Referencing time as zulu (presented on a 24-hour scale marked with Z) assures there are no misunderstandings as to actual time meant.

Requirements for a Traffic Flow Management Serious Game

For a serious game to be successful in aiding traffic flow management, it must meet multiple requirements. Many of these requisites involve creating a game environment that faithfully simulates the actual world in which TMCs work. The game needs to accurately represent the resources in the NAS, including airports, routes, fixes, sectors, and facilities. The dynamics of the different aircraft types must also be accurately represented, and flight behavior must reflect real flights (e.g., filed flight levels, standard speeds, and structured routings). The game has to provide realistic depictions of current and forecasted weather, similar to the CIWS displays used in operation. The assessments of capacity impacts have to be accurate; aircraft holding and other signs of demand and capacity imbalances should correctly reflect the simulated weather conditions.

The game environment must also present information in ways that replicate the systems and displays that TMCs use. The game needs to be able to emulate the Traffic Situation Display and Flight Schedule Monitor to allow TMCs to assess demand, and it needs to emulate CIWS and CoSPA to allow TMCs to assess capacity. Useful graphical user interface behaviors, such as filtering flights, zooming in and out, and specifying airport resources, must also be replicated. When the game is being used to test the impacts of new information on decision making and operational outcomes, it must support the incorporation of new components.

An effective TFM game will provide trainees with an experience that is true to what they will encounter on the job. The scenarios in the game must replicate the complexities of an actual operational day, including information uncertainty and multiple overlapping decisions. To support the rapid incorporation of lessons learned into operations and to enable the creation of a large and varied library of experiences for trainees, the game must utilize automation for reducing the time and effort needed to create new scenarios. The decision choices offered in the game should be representative of decisions that TMCs would actually make and may make in the future. The means to address demand and capacity imbalance should be based on the same choices that TMCs have now—airspace flow programs (AFPs), ground delay programs (GDPs), ground stops, and reroutes. These decisions should be offered at different times throughout the scenario day to replicate the conditions under which controllers currently operate. If the game is to be used for evaluating new procedures, it should provide information on the procedure's potential impact, reflect the decision process, and model the resulting outcomes.

A useful game will meet several functional requirements. The game should allow a scenario day to be simulated in only a few minutes. The game player needs to be able to replay and make different decisions for the same "day" so that he or she can view and assess the varying outcomes of decisions. The game should also provide some objective feedback about the system's performance, given the decisions made during the game. These metrics or scores should reflect operational metrics (e.g., delays) that are used currently to diagnose NAS issues or experimental metrics (e.g., number of times that a traffic manager views information to make a particular decision) that may provide new insights into operations. Ideally, the game is web-based to allow access for players who may have limited alternative access to the final game product.

Development of the NASPlay Serious Game Prototype

While a serious gaming architecture could support the multiple applications described in the previous section, Lincoln Laboratory pursued the following development goal for NASPlay: develop a serious game architecture that supports ingesting data from actual operational days and provides the game player with a choice of alternative traffic management initiatives that result in an operationally relevant score for each alternative.

Figure 5 illustrates the architecture and assignment of functional capabilities in the current NASPlay system. There are three major components, each with specific and well-defined functional capabilities and interfaces: the NAS simulation engine (NSE), the NASPlay game server, and the game interface. Trainees would interact directly with the game interface.

The computational performance of the simulation is insufficient to run in real time while the game is being played. To accommodate the realities of current simulation performance limitations, the NASPlay developers formulated a constrained-choice concept of operations

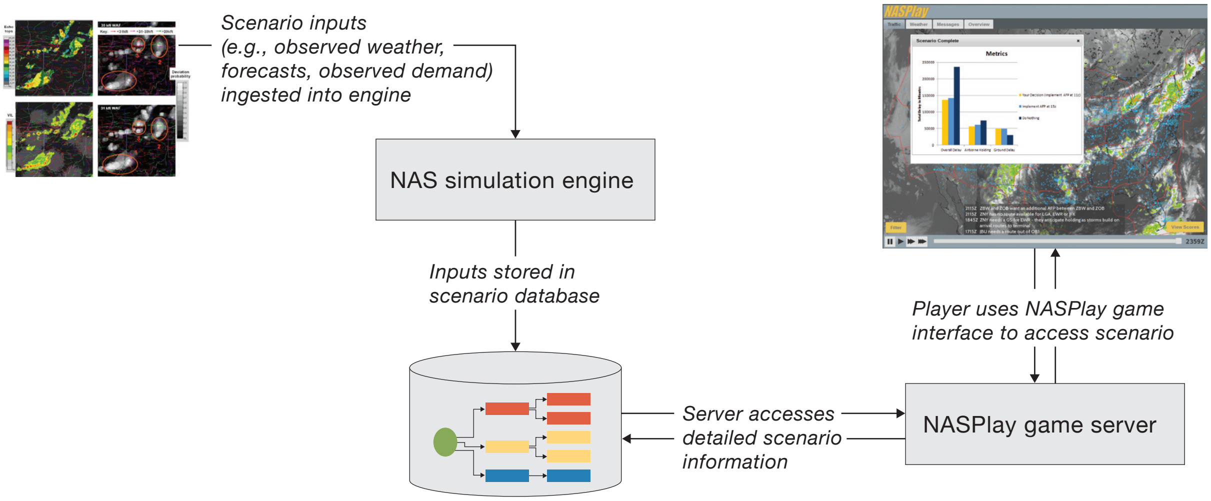


FIGURE 5. The NASPlay system architecture contains scenario inputs, the NAS simulation engine, NASPlay game server, a scenario database, and the NASPlay game interface. The scenario inputs are ingested into the NAS simulation engine and stored in a scenario database. When the scenario is accessed by the NASPlay game server (through the NASPlay game interface by a player), the database is accessed for the detailed scenario information, including the action choices indicated by the colors in each branch above.

in which a limited set of traffic management decisions is made available to the game player at a sequence of discrete decision points during gameplay. As a result, the NAS simulation is decoupled from the actual gameplay, and changes in NAS states that result from players' decisions are precalculated, stored on disk, and provided to the player by the server.

The use of the constrained-choice concept has several implications. Clearly, fewer decision options are available to the game player, and the extent of option exploration is limited to those that were considered by the author of the scenario. However, it is still possible to include a huge number of choices that reflect all the decisions that a TMC could realistically make. The specification of an explicit set of decision options also makes it possible to make clear comparisons between different traffic management strategies and choices. The decoupling of the game from the simulation also makes it possible to provide a large library of game scenarios that may be accessed as part of a progressive training regimen or a concept-engineering and validation exercise. This approach puts much less demand on the network and server, allowing virtually any number of players to access the game at once. With some forethought in scenario development, the constrained-choice concept can provide a rich and varied environment to address many of the challenges in evaluating and training traffic management planning and decision making.

NAS Simulation Engine

The NSE implements the rules for NAS behavior (e.g., its response to external inputs such as scheduled demand and weather impacts) that define specific gameplay scenarios. The NSE schedules flight departures, models flight trajectories, and implements default behaviors of the NAS in response to external events or conditions that arise as the simulation proceeds (e.g., what happens to flights entering an air traffic control [ATC] sector when that sector is at capacity). The NSE also provides the capability to model commonly used traffic management initiatives, such as ground delay programs, on the basis of forecast traffic demand and constraints. The NSE provides the capability to harvest data about the evolving state of the NAS during the simulation (e.g., the location of flights and their delay status) and calculates NAS-wide and local performance metrics. Finally, the NSE provides fast-time simulation capabilities to facilitate the generation of NAS outputs corresponding to each branch of the constrained-choice decision tree.

To create the level of realism required for meaningful game development, the NSE must provide fine-grained control of flight trajectories and air traffic control actions, particularly in response to external events such as thunderstorms. The NSE must provide a way to extend or replace default behaviors (e.g., pilot decisions to accept or reject routes through convective weather-impacted airspace or the response of ATC to weather impacts) with models such as the Convective Weather Avoidance Model for pilot decision making in convective weather–impacted airspace, or the Controller Workload Model for sector capacity.

After a technical evaluation of an array of existing simulation products, AirTOpsoft's simulator, AirTOp [6], was chosen because of its agent-based foundation and flexible development environment. AirTOp's implementation enables fine-grained control over several key elements of NAS operation and simulation:

- Dynamic capacity constraints. Simulations may be initialized with time-varying capacity constraints on any airspace resource that is defined in the NAS adaptation.
- Options for tactical weather avoidance. AirTOp provides mechanisms to implement tactical weather avoidance options, such as no-notice holding and trajectory vectoring to avoid weather. In addition, thresholds (often referred to as hooks) can be set to trigger diversions, ground stops, and other tactical responses to airspace constraints.
- Hooks for calculation of default and custom performance and scoring metrics. AirTOp supports the specification of software watch points that can trigger data analysis and the output of user-specified simulation state data for incorporating the generation of performance and scoring metrics into the simulation.
- Data are stored in easily modified text files.

After the development staff spent several weeks familiarizing themselves with the AirTOp environment, they input the baseline structure for the NAS (e.g., Air Route Traffic Control Center boundaries, ATC sector boundaries, navigation fixes, jet routes, aircraft types, and airports) into the simulation. Data for a full day's flight plan schedule were assembled and input to AirTOp. As is common with navigation data, a significant amount of "cleaning" of the data was required to return reasonable output:

- Correction (where possible) or removal of ambiguously or incorrectly specified navigation fixes from flight plans
- Assignment of aircraft performance statistics when the aircraft type is unknown to AirTOp
- Proper sequencing of departure times to ensure temporal continuity of flight plans that have multiple stops and continuation legs
- Filtering of flight plans that are outside the scope of the game scenario to reduce simulation run time
- Specification of realistic cruise altitudes and air speeds for flight plans that are missing this information
- Conversion of scheduled flight plans into AirTOp's input format
- Determination of which entry is most accurate if the same flight plan appears multiple times in the data; removal of any "loops" from routings
- Conversion of units of measurement, especially for speed (e.g., Mach, indicated airspeed, true airspeed)

Five critical weather-impact capabilities were also implemented in NASPlay: time-varying winds for flight trajectory modeling; time-varying air traffic control sector capacity constraints that include considerations for convective weather impacts; time-varying air traffic flow capacity constraints that account for convective weather impacts; time-varying airport capacity; and time-varying fix capacity.

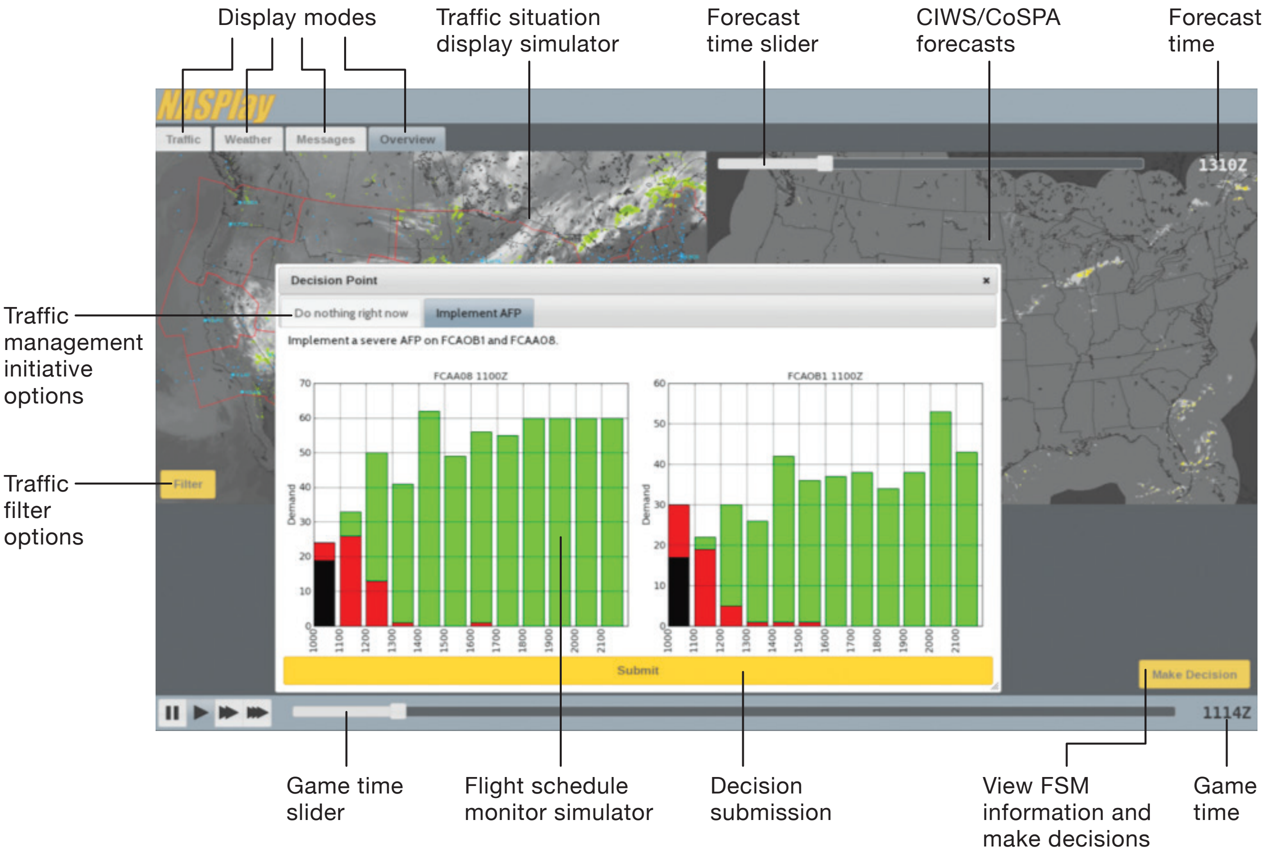
In addition, several initial performance metrics were implemented: individual flight delay, separated into ground and airborne portions, as well as planned and unplanned portions; hourly measurement of aggregate delay and holding; time of flight; fuel burned; and cancellations and diversions.

NASPlay Game Server

The primary role of the NASPlay game server is to provide to the game client the current state of the NAS resulting from the player's decision choices up to that point. The NAS state includes the flight plans and locations of all flights, outputs from operational models for the current weather and weather forecasts, and stakeholder comments or tactical responses (e.g., a request for a ground stop or diversion) derived from external sources (e.g., the National Traffic Management Log) or automatically generated by the NSE during scenario preparation. The server also records player decisions and interactions that will be used for postgame analysis.

Game Client

The game client is the player's window on the NAS world. It renders the game display that provides (1) the current state of the NAS, such as flight locations and plans, current NAS performance statistics, and emulation of commonly used tools such as the Flight Status Monitor; (2) feedback



Decision Point

Do nothing right now | Implement AFP

Implement a severe AFP on FCAOB1 and FCAA08.

Submit

FIGURE 6. The NASPlay user interface includes display modes, time sliders, filtering options, game time, and other data critical to the decisions required, such as the Flight Schedule Monitor.

comments from other NAS stakeholders; and (3) external factors, such as the weather. It provides the game clock control, allowing the player to start and pause the action or rewind to review the previous state of the world. The client prompts the player for decisions, providing NAS modeling information relevant to the decision options. Finally, the client passes selected player interactions to the server for logging and postgame analysis. The client in the current NASPlay prototype is shown in Figure 6.

Game Scenario

The development of the game scenario is key to the success of the game. Gaining TMCs' acceptance of a traffic management serious game would be impossible if the scenario did not capture the complexities of traffic management and the subtleties of the operational environment. To this end, the NASPlay developers chose a particularly impactful day that had questionable traffic management decisions made during the operation. A similarly impactful weather day was further explored with respect to forecast uncertainty and decision making [7].

On 11 September 2013, a group of severe thunderstorms developed between Maine and Tennessee around 1600Z (4 p.m.), impacting eastbound arrivals starting around 1700Z. Traffic managers at the Command Center in Virginia opted to address the capacity constraint imposed by these storms by rerouting New York–bound flights from Fort Worth, Houston, and Memphis centers south through the Vulcan Playbook[2] (VUZ) reroute and AZEZU Playbook reroutes, and by tactically managing traffic through ground stops. Managers also implemented Airspace Flow Programs at 1650Z for two flow-control areas (shown in Figure 7) from 1915Z and 1945Z. There was significant NAS disruption, including 69 diversions,

[2] A playbook contains a set of standard routes that ATC can utilize to fit a particular set of circumstances when the preferred routes are not available. These routes were created to allow for rapid implementation of rerouting as needed.

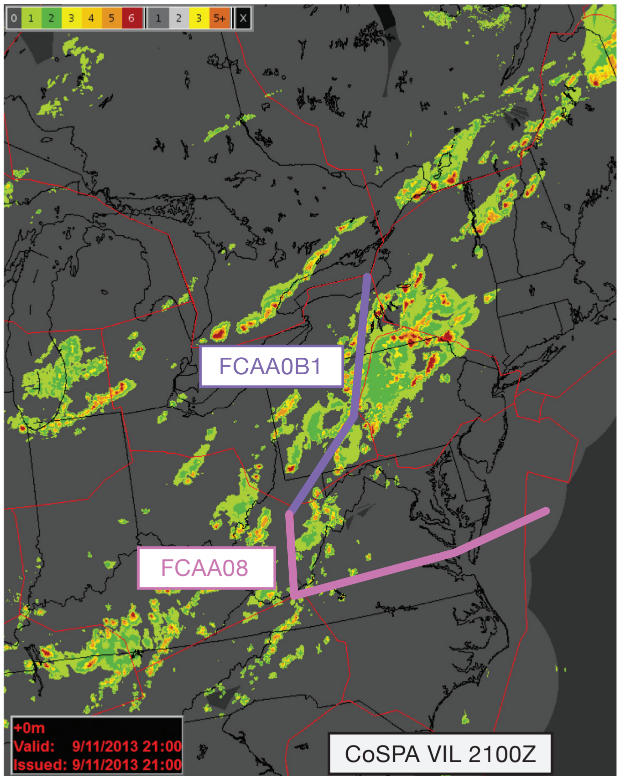


FIGURE 7. The map shows the flow control areas (FCAs)—FCAAOB1 and FCAA08—that were restricted and the convective weather existing on 11 September 2013 at 21:00 zulu. Data are from the CoSPA system, which uses vertically integrated liquid measurements to predict convective weather activity.

55 holding events (totaling 21 hours), 13 ground stops (totaling 7 hours), and 72 taxi-backs to the airport gates.

The critical aspects of a scenario for game reconstruction include the timeframe and area of interest, the decisions available, the information available, and the metrics by which the decisions are evaluated.

The area of interest for this scenario is the New York (ZNY)–District of Columbia (ZDC) area. Thus, it is important for the player to be able to view and filter flights arriving and departing this area of the NAS. Also important is the player's ability to zoom into this area to discriminate the local weather and traffic. The timeframe of interest is from 1700Z to 2100Z, a busy period during which the weather significantly affects the high-demand traffic areas. For decision-making purposes, it is important to be able to view not only the unfolding weather and traffic but also the forecasts of weather and traffic demand for this timeframe.

It is critical to identify the key decisions available to the TMC to address the demand and capacity imbalance issues for a particular scenario. The strategic traffic management decisions available to the TMCs for this scenario include reroutes, AFPs, GDPs, and ground stops. For convective weather impacts in ZNY and ZDC, the appropriate AFPs include FCAOB1 (the eastern boundary of Cleveland center) and FCAA08 (west/east line through Washington center). Some of the decisions must be made no later than four hours before the expected impact in order to have the desired effect on the traffic. Thus, forecasts for the 1800Z timeframe and beyond need to be available to the game player by no later than 1400Z. GDPs for the New York airports were made available as potential decisions as well.

An example constrained decision tree was created for the game scenario by using these key decisions. Figure 8 illustrates a traffic management initiative decision tree that enumerates the set of possible decisions for this scenario. At 1315Z and 1715Z, the player is able to choose whether to implement an AFP or a reroute and whether the AFP should be "mild" or "severe." If a reroute is chosen, then the player is also offered the choice at 1915Z to implement a GDP or not. Seven outcomes for this scenario are possible, and the possible decisions in this example are constrained. A full scenario would allow for the 10 to 100 choices that an actual national air traffic manager would experience in a convective weather day.

To adequately represent the scenario in a context familiar to the game player, the information presented must be consistent with that used by a TMC. Two information sources are used to assess demand over time: the Traffic Situation Display and the Flight Schedule Monitor, both shown in Figure 3. To assess capacity, TMCs must have adequate knowledge of the current and forecasted location and severity of the weather, such as is depicted in the CIWS and the CoSPA tools shown in Figure 4. Strategic TMCs also receive input from local air traffic control facilities (ARTCCs[3], TRACONs[4], and towers), as well as from their airline customers, about what decisions to implement

[3] Air Route Traffic Control Centers handle primarily en route aircraft on instrumented flight plans; 21 centers cover the regions over the United States.

[4] Terminal Radar Approach Control facilities handle ATC operations near major airports, primarily aircraft arrivals and departures.

via phone calls. To simulate this communication in the game environment, a chat window was implemented to allow facility and airline agents to provide their opinions on the decision options. Chat messages were derived from the National Traffic Management Log and created in response to simulated events.

To understand whether a game player's decision was "good" or "bad," operational performance metrics must be established. A common metric used by the ATC community is the amount of delay accrued during an event for the NAS. Additional metrics—airborne holding time, ground delay time, uncontrolled delay, fuel burn, and number of diversions and cancellations—were identified to indicate the quality of a decision. Filters identifying where and when the delays occurred also provide an indication of how the traffic was affected by decisions. The defining and weighting of performance metrics in scoring are areas of active research, and these are expected to evolve significantly as NASPlay development continues.

To acquire the data to ensure the scenario fidelity for the game, the following required data from 11 September 2013 were assembled:

- NAS definition data
- Scheduled traffic data
- Wind data
- Lincoln Laboratory's CIWS and CoSPA weather data archives for the NAS
- Lincoln Laboratory's Route Availability Planning Tool data for fix capacities
- Lincoln Laboratory's Traffic Flow Impact data for sector and flow capacities
- Command center teleconference and National Traffic Management Log data (what decisions were considered when, inputs by ATC facilities and airlines)

Emulations of the Traffic Situation Display, Flight Schedule Monitor, CIWS, and CoSPA were created to ensure realistic representation of the information consistent with the traffic management context.

Validation and Evaluation of NASPlay

Both validation and evaluation are required to ensure that NASPlay meets the needs of the NAS users. A detailed report of the validation for NASPlay is provided in Davison Reynolds, DeLaura, and Soulliard [8]. It

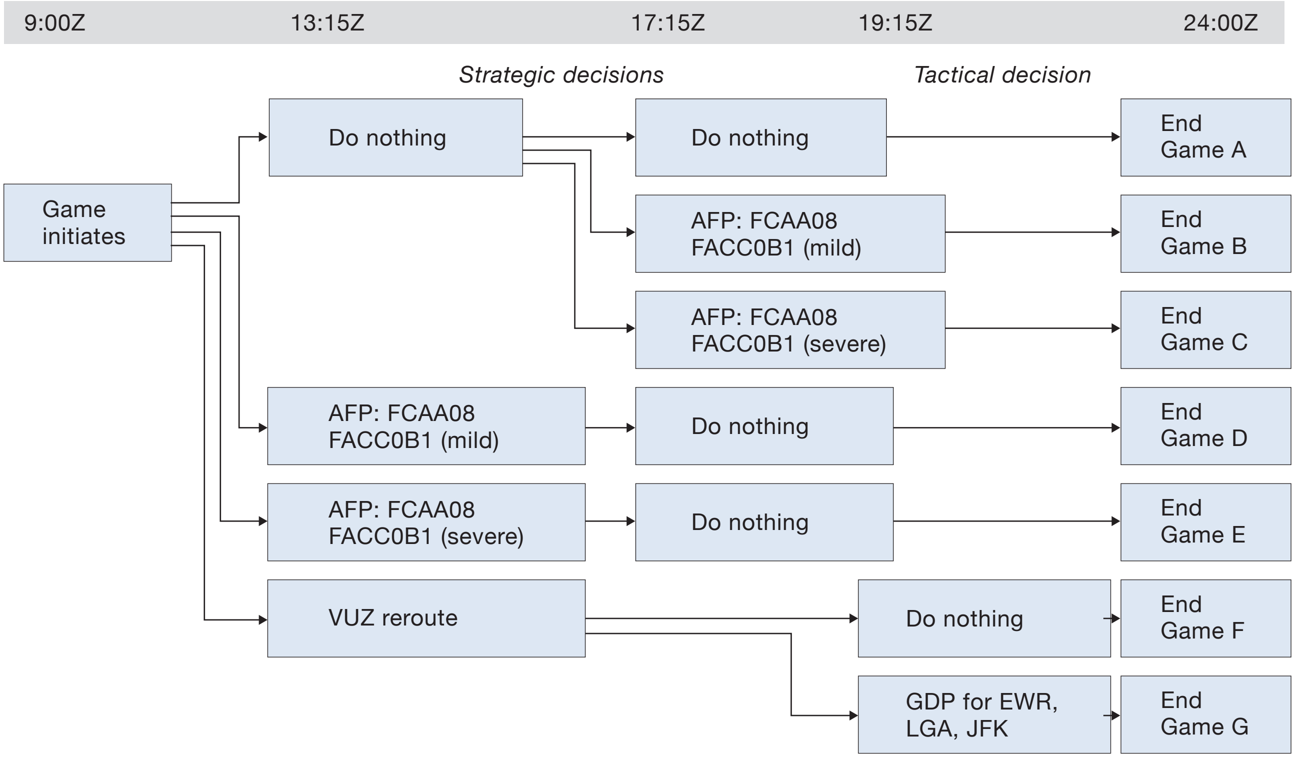


FIGURE 8. This simplified decision tree for a constrained choice game illustrates the choices available to the player at different times during the scenario.

is important that the simulation dynamics, the traffic demand, and the weather capacity algorithms all work in concert to provide a valid representation of the NAS because conclusions drawn from an invalid model would not transfer to the real NAS. Likewise, if the NAS users evaluate NASPlay and find that it does not represent the NAS in some critical way, NASPlay will not be accepted within the community. Thus, initial validation and evaluation have been attempted for NASPlay.

An initial validation was performed for a nominal, unconstrained (fair weather) operational scenario, taken from operations on 21 October 2012. The primary focus of the validation was to ensure that the schedule cleansing and wind data ingest resulted in reasonable flight plans, trajectories, and overall number of operations.

Individual Flight-Level Validation—Flight Simulations

The total number of flights flown by the simulation was 34,928, a number roughly in line with the FAA's OPSNET[5] ASPM[6] 77 terminals' count of approximately 56,000 operations for the day. Note that ASPM operations include both arrivals and departures for domestic airports, so their count is roughly double the number of flights for 21 October 2012. However, the ASPM count does not include general-aviation flights.

AirTOp time of flight was compared to observed time of flight for each scheduled flight with a corresponding observed departure. The results of the comparisons for flights between the 34 largest airports in the continental United States are presented in Figure 9 and show good agreement between simulated and observed flight times. Top-down map views of several flight plans for different origin-destination pairs were inspected to ensure that "doctored" simulation flight plans were reasonable. The distribution of flight altitudes as a function of flight distance was also examined to confirm that cruise altitudes were sensible in NASPlay. Finally, an initial performance measurement analysis capability that will form the basis for the game scores was developed. The capability currently assesses ground delays (planned and unplanned), airborne delays, cancellations, and diversions, all adjusted for the types of flights impacted (e.g., passenger, cargo, or general aviation).

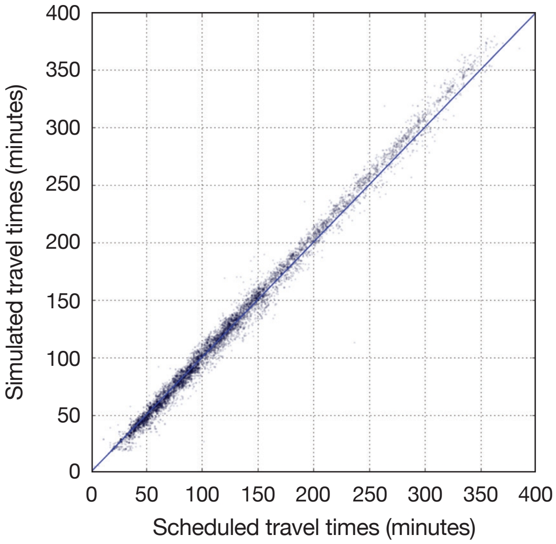


FIGURE 9. A comparison of simulated and observed flight times for major airport origin-destination pairs are shown.

Flow-Level Validation—Capacity Modeling

A time-variant capacity constraint was developed for flows and sectors. Each flow captures flights going through its area in a certain direction. The algorithm was previously developed and verified at Lincoln Laboratory. Testing indicates that the simulation is performing as the model predicts.

NASPlay Evaluation

The NASPlay prototype was initially evaluated by several NAS user groups, including trainers and traffic management specialists from the Air Traffic Control System Command Center (ATCSCC), the manager of tactical operations in the Northeast United States, former en route/TRACON/tower controllers, and representatives from two airlines. An initial introduction to the prototype, which included the potential concept of operations and use for the tool, was provided to the evaluators. Users then played through the demonstration scenario, seeking out diagnostic information and making their own choices. Once the users completed the demonstration, they were asked what, if any, value the prototype concept would have in their jobs and what information or functionality is missing to achieve that value. Table 1 itemizes the

[5] Operations Network is the official source of data on NAS traffic operations and delays.

[6] Aviation System Performance Metrics is an online database of information on flights to and from the 77 U.S. airports.

Table 1. NAS Users' Estimation of the Value of NASPlay to Their Operations

USER GROUP	VALUE OF NASPLAY TO OPERATION
ATCSCC trainers	Integration of NASPlay with their laboratory training environment to produce fast-time "what if" decisions to a set of specified scenarios
ATCSCC traffic management specialists	Ability to conduct over-the-shoulder, on-the-job training with new traffic management specialists; ability to better understand the interaction of traffic management initiatives with one another in a controlled environment
Manager of tactical operations in the Northeast United States	Capability to support continual offline demand and capacity imbalance identification; evaluation of traffic management decisions with objective metrics
Former air traffic controllers	Ability to try out and evaluate the effects of new procedures offline; training in severe weather decision making
Airline representatives	Ability to model and better understand the effect of traffic management initiatives on their businesses; this understanding could lead to effective lobbying for particular initiatives on strategic planning

Table 2. NAS Users' Suggested Improvements for NASPlay

USER GROUP	SUGGESTED IMPROVEMENT
ATCSCC trainers	Generate many severe weather scenarios; explore connecting NASPlay to training Flight Schedule Monitor
ATCSCC traffic management specialists	Incorporate airline cancellations and pilot diversions into functionality; make NASPlay multiplayer and web-based for a single scenario
Manager of tactical operations in the Northeast United States	Generate tactical scenarios for NASPlay focusing on a single en route center and/or TRACON; provide ability to continuously monitor airport surface status; make NASPlay scoring consistent with the FAA's internal AERO operational evaluation statistics page
Former air traffic controllers	Incorporate airline cancellations, pilot diversions, and tactical rerouting into functionality
Airline representatives	Make the prototype available for airline use

specific value to their jobs that the user groups saw for the prototype.

The users also offered suggestions for additional information and functionality to improve the ability of NASPlay to meet their identified needs; Table 2 itemizes these suggestions.

Future Development

The NASPlay prototype was developed to address serious shortfalls in current FAA capabilities for training air traffic managers, evaluating current and proposed NAS operational procedures, and developing and validating new operational concepts. Its platform integrates a commercial simulation capability with both the Laboratory's novel algorithms for severe weather capacity and its gaming interface. The prototype's output was validated in both fair and severe weather.

All the NAS users who evaluated NASPlay's operational value and functionality prioritization saw useful applications of NASPlay for their respective jobs. Many of the suggested improvements in information and functionality are possible and desirable to accomplish within the next year. A more detailed user evaluation is planned for the end of next year. The goal of that evaluation will be to gather input about both the available decision choices in the assembled scenarios and the usability of the current NASPlay prototype.

Over the next year, NASPlay will be expanded into the tactical traffic management realm, with the development of tactical scenarios (regionally focused rather than nationally focused). Additional possibilities for NASPlay's expansion include

- Multimodal performance scoring to evaluate decision making according to alternative performance criteria, such as environmental impact or passenger experience
- Multiplayer gaming for advanced training and evaluation of future concepts, such as dynamically configured airspace
- Agent-based Monte Carlo simulations to realistically assess the potential benefits of new forecast tools and procedures, accounting for limitations, such as forecast accuracy and uncertainty in the response of pilots and controllers to events
- Real-time simultaneous simulations and scoring of potential alternate outcomes to guide planners in operational decision making ■

References

1. Federal Aviation Administration, National Severe Weather Playbook, 2016, https://www.fly.faa.gov/PLAYBOOK/pbindex.html.
2. D.L. Klingle-Wilson and J. Evans, "Description of the Corridor Integrated Weather System (CIWS) Weather Products," MIT Lincoln Laboratory, Lexington, Mass., Project Report ATC-317, 1 August 2005.
3. M.M. Wolfson, W.J. Dupree, R. Rasmussen, M. Steiner, S. Benjamin, and S. Weygandt, "Consolidated Storm Prediction for Aviation (CoSPA)," 13th Conference on Aviation, Range, and Aerospace Meteorology, New Orleans, La., 2008, http://www.ll.mit.edu/mission/aviation/publications/publication-files/ms-papers/Wolfson_2008_ARAM_MS-30236_WW-14159.pdf.
4. D.L. Klingle-Wilson, "Integrated Terminal Weather System (ITWS) Demonstration and Validation Operational Test and Evaluation," MIT Lincoln Laboratory, Lexington, Mass., Project Report ATC-234, 13 April 1995.
5. M. Robinson, R. DeLaura, and N. Underhill, "The Route Availability Planning Tool (RAPT): Evaluation of Departure Management Decision Support in New York During the 2008 Convective Weather Season," Eighth USA/Europe Air Traffic Management Research and Development Seminar, Napa, Calif., 2009, http://www.ll.mit.edu/mission/aviation/publications/publication-files/ms-papers/Robinson_2009_ATM_WW-16318.pdf.
6. AirTOpsoft, AirTOp fast time simulator, 2016, http://www.airtopsoft.com/.
7. H.J. Davison Reynolds, R. DeLaura, J. Venuti, and M.M Wolfson, "Uncertainty and Decision Making in Air Traffic Management," AIAA Aviation, Technology, Integration, and Operations Conference, Los Angeles, 12–14 August 2013.
8. H.J. Davison Reynolds, R. DeLaura, and B. Soulliard, "How Good Is Good Enough? Exploring Validation for an Air Traffic Control Serious Game" chapter in *Advances in Human Factors, Business Management, Training and Education: Proceedings of the Applied Human Factors and Ergonomics 2016 Conference on Human Factors, Business Management, and Society*, J.I. Kantola, T.Barath, N. Salman, and T. Andree, eds. Switzerland: Springer International Publishing, 2017.

About the Authors

Hayley J. Reynolds is an assistant leader of the Informatics and Decision Support Group. Her expertise is in human-systems integration and systems engineering. Since joining Lincoln Laboratory in 2009, she has worked extensively on the design and development of aviation decision support systems, biosurveillance systems and emergency management systems. She also leads several programs centered on counter-human trafficking. She holds a bachelor's degree in psychology from the University of Illinois at

Urbana-Champaign, and a master's degree in aeronautics and astronautics and a doctoral degree in aeronautical systems and applied psychology, both from MIT.

Brian C. Soulliard is currently a staff member at Google. While at Lincoln Laboratory, he was the lead software developer for NASPlay. He graduated with a bachelor's degree in software engineering from the Rochester Institute of Technology with minors in computer science and in game design and development.

Richard A. DeLaura is a technical staff member in the Air Traffic Control Systems Group. Prior to joining Lincoln Laboratory in 2000, he was a research scientist at the University of Massachusetts Dartmouth, developing software and curriculum to introduce advanced mathematical concepts to middle and high school students. He has authored or coauthored several conference proceedings and Federal Aviation Administration research reports. He holds a bachelor's degree in chemistry and physics from Harvard University.

Rapid-Play Serious Games for Technology Triage

Robert M. Seater

Rapid-play serious games can allow players to gain intuition about the use of a proposed capability, enable researchers to examine that capability's influence on tactics and procedures, and collect quantitative data that supplement qualitative user feedback to inform decisions about which new technologies should be pursued with future development.

The analysis of user-facing future technology is a difficult task but one that plays an important role in the process of research, development, and technology evaluation (RDTE). The RDTE process includes many facets, ranging from brainstorming potential threats and opportunities all the way to prototyping and conducting field evaluations. An efficient RDTE process is important to avoid missing opportunities (culling good ideas) or investing too much effort into dead ends (failing to cull bad ideas). Unfortunately, many technology programs fail before they even get started because they are seeking to provide a capability that users do not need or will not accept. However, recognizing which technologies will be useful before they have been developed, prototyped, and field tested can appear to be a chicken-and-egg problem—how can we triage a set of capabilities before they exist?

To understand how to address this problem, it is first important to articulate what makes the task difficult. Consider, for example, a proposal for a novel detection technology that is light enough to be used as a wearable sensor for infantry squads. If it is our job to decide if that technology is worth maturing for that application, we face several immediate challenges:

- First of all, because the technology does not exist yet, we don't know what technical trade-offs it will be able to offer, what technical specifications we would want it to meet, or where additional research is most needed to close the gap. Is it more important that the sensor have a low false-positive rate or a high range? A high-fidelity image or a fast update rate? We don't even know where a research program should focus its efforts or if the end result will be acceptable to users.

- To answer such questions, one typically turns to current domain experts and users. Involving experts and users can provide valuable feedback on the utility of the new capability and its likelihood of being accepted. So, we might ask current squad soldiers what they would find most helpful in a wearable sensor. Unfortunately, most expert decision makers are intuitive thinkers used to dealing with concrete situations, not abstract thinkers who have a theoretical formalism that can generalize to future scenarios [1]. Expert users may not understand why they are experts and thus not understand what new capabilities will help them in a novel (future) environment [2].
- To make the problem more concrete for the domain experts, we might run a tabletop exercise or seminar-style wargame [3] so that they can get some intuition for what it is like to use the proposed capability and how it might change their operating environment. However, after such an exercise (or even a few), the domain users are still novices at using the new technology, and they haven't had much chance to experiment with how to use the technology in different ways or to explore how it might change doctrine and best practice. The squad members have only had a couple of chances to experience how a wearable sensor might change their behavior and how to incorporate it into current doctrine. In an adversarial setting, the red force will also not have had time to develop exploits and counter-tactics. Furthermore, we still rely on participants' qualitative descriptions of what they liked or didn't like about using the sensor—a method hindered by users with dominant personalities or experts who are not good at theorizing.
- To address the issues that come from a small number of qualitative data points, we might run a large number of exercises and instrument users to collect data on their performance and behaviors. However, that is an expensive proposition if one uses traditional exercises and tabletop scenarios that take hours or days to run, that pull experts away from other tasks, and that require participants to travel to a common location. Such an approach is costly, burdensome, and slow. The early phases of RDTE can seldom afford any of those drawbacks, and developers usually face pressure to provide a quick, cheap, and low-burden estimate of where to focus subsequent efforts so that the next phase of the program can get underway with most of its budget intact. If we spend all our time understanding what wearable sensor to build, the program may be canceled or the problem may simply become obsolete as the world changes.

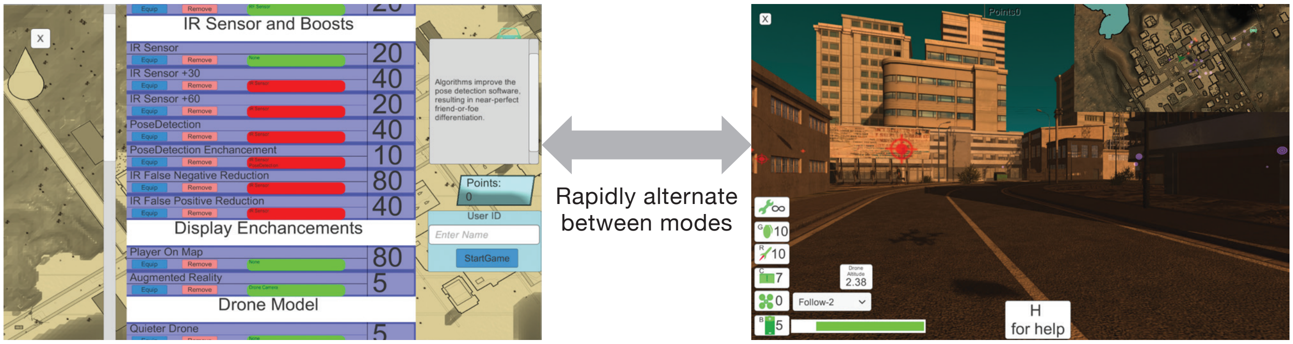
So what we are looking for is a method of providing users with a concrete environment in which they can explore a future capability many times to build intuition, collect both quantitative and qualitative data on their performance and preferences, and do so without consuming a lot of program time, participant time, or budget.

HIVELET: Crowdsourcing Human Creativity

For the last few years, MIT Lincoln Laboratory has been using serious games to aid in technology assessment programs. One of the most recent efforts is the Human-Interactive Virtual Exploration for Low-Burden Evaluation of Technologies (HIVELET). The HIVELET approach focuses on early RDTE, especially when suites of emerging technology are being considered for user-facing roles. This approach combines economic game theory [4] with rapid-play digital simulations to collect quantitative data, improve qualitative feedback, and crowdsource the ingenuity of human experts.

Under the HIVELET approach, players alternate between two modes—capability selection and mission simulation, as illustrated in Figure 1.

- Capability selection allows players freedom to select different combinations of conceived capabilities, allowing them to formulate and explore different strategies that may deviate from current doctrine. However, the selection mode prevents a player from simply choosing all available capabilities; they must manage a limited budget (representing cost or weight), forcing them to think critically about what capabilities they really need and to carefully prioritize the available capabilities. Players are not only judging if a capability is useful but also if it is useful enough, given its drawbacks and alternatives.
- Mission simulation gives players a chance to try out the set of capabilities they selected to get feedback about effectiveness and to build intuition about what did or did not work well. The mission simulation is focused on being short (e.g., minutes not hours) so that players can make multiple attempts within a single sitting to



Algorithms improve the pose detection software, resulting in near-perfect friend-or-foe differentiation.

FIGURE 1. Under the HIVELET approach, players alternate between two modes—capability selection (left) and mission simulation (right). The depicted capability list shows unmanned aerial vehicle (UAV)–mounted sensor capabilities and upgrades the player can mix and match, each with an abstract resource cost. The depicted mission simulation is a first-person, three-dimensional simulation of an urban environment.

explore different strategies and build more intuition through iteration. To achieve these objectives, the mission simulator captures a key aspect of a critical decision point in the real world and abstracts away details not relevant to the evaluation at hand. Design principles and scoring incentives are used to create an environment that accurately recreates the pressures of the real world while simplifying the real-world simulation enough to shorten the duration of gameplay.

After completing the mission simulation using the selected capabilities, players return to the selection mode. They can stick with their prior choices, refine their strategy, or try an entirely different approach. They then repeat the simulation, continuing to alternate back and forth between the two modes. The alternation forces players to combine abstract thinking about the value of various capability combinations with concrete feedback and intuition about the use of those capabilities on a mission. Data collected during the game reveal players' preferences, behaviors, and performance and can be used in researchers' quantitative analyses that complement the qualitative feedback provided by participants. With appropriate design of the framework, a participant can complete several cycles of selection and simulation in an hour.

Both portions of the game can be hosted online and played remotely by participants, thereby greatly reducing the burden and cost per each data point. A wide range of players remotely playing a series of short simulations can quickly compile a lot of data that can shed light on the trade-offs and priorities for the capabilities being modeled. Researchers can also vary the mission parameters to see how players change their preferences and strategies, thereby providing insight into the application or the concept of operations (CONOPS) for which a given future capability is likely to be best suited. For example, the infantry mission simulator shown in Figure 2 can be run using a range of different terrain types and mission objectives to determine the flexibility or specialization of certain capabilities.

This approach is a form of crowdsourcing—using humans in large numbers to perform tasks that are difficult to automate. In this case, the task being automated is the creative thinking and ingenuity about how to mix and match future capabilities of various quality levels into a coherent and effective strategy that manages the risks presented by a real-world mission situation. Humans are not good at fine-tuned optimization, but they are excellent at creatively finding good combinations from within a very large decision space. This approach is thus well suited to the early stages of RDTE, in which we need to rapidly triage an enormous design space to focus more systematic traditional evaluation methods on the most promising options. HIVELET isn't the end of the RDTE story, but it can be a critical step in making other techniques more focused, more efficient, and ultimately more likely to succeed than they would be if used alone.

Application to Infantry Technologies

The HIVELET technique has been used to evaluate how a small unmanned aerial vehicle (UAV) integrated into tactical infantry missions might fundamentally change how such squads operate. The game modeled 29 capabilities (e.g., sensors and control mechanisms) and capability

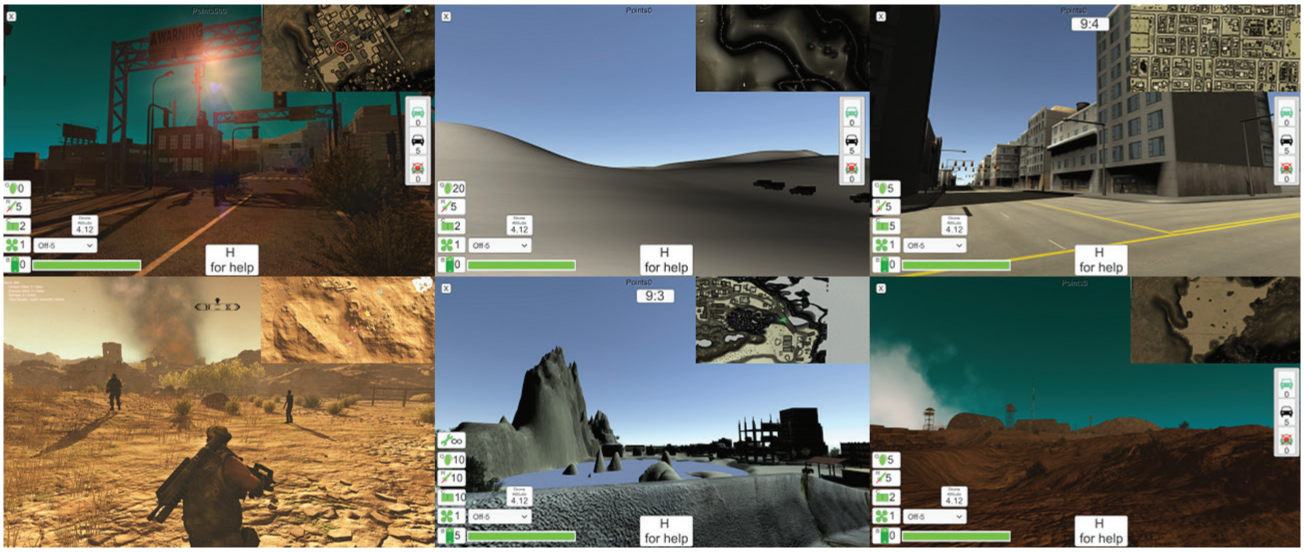


FIGURE 2. A player executes a tactical infantry mission in a digital simulation, using in-game models of concept technologies. Domain experts who rely on experience and intuition often find it easier to provide feedback on concepts when they can try them out in a simple simulation rather than when they are asked to engage in a purely theoretical discussion. Researchers can examine how player behavior and preferences change in different environments and for different missions. The environments shown here, left to right, are a ruined city, an arctic tundra, a large city, a rocky desert, an island, and a night mission.

upgrades (e.g., enhancements to the sensor quality or to the player's weaponry). In the mission simulator, players navigated a three-dimensional (3D) real-time environment and attempted to recover data from a predator or reaper drone that went down in a hostile urban environment (Figure 3). The player has to balance finding the objective quickly with safely navigating the terrain to avoid or neutralize threats.

The simulated city covered several blocks totaling about half a square mile of dense urban terrain. Within the city were randomly clustered groups of 20 to 50 civilians and 10 to 20 dismounted hostile soldiers on the streets and in alleys. Civilians and hostiles varied their behavior between standing, walking, investigating noise, and fleeing from noise. Once alerted by noise, hostiles became more alert, and civilians had a chance to flee or cower. Civilians and hostiles were dressed in a similar fashion, and some hostiles were dressed identically to civilians. Only hostiles were armed. The downed target would be randomly placed at ground level somewhere within the map bounds. There were between 0 and 2 false positives for the radio-frequency (RF) signal of the target and between 0 and 10 false positives for infrared (IR) signatures for people. Future capability upgrades would differentiate those false positives and more accurately classify targets.

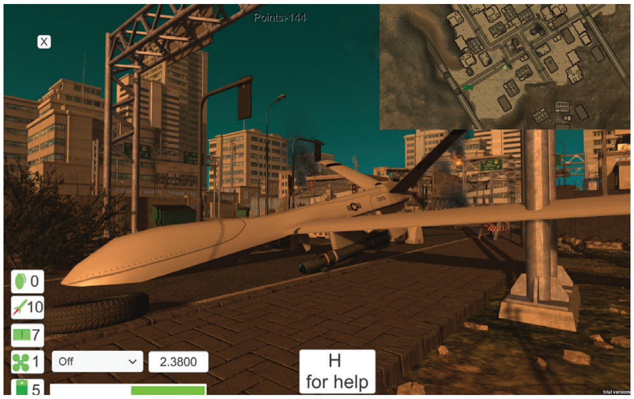
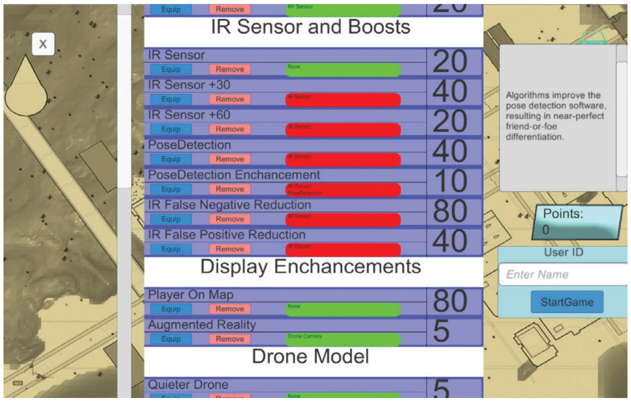


FIGURE 3. In this mission simulator, players must navigate a hostile urban environment to find a crashed predator or reaper drone, recover its data, and extract those data safely. They must choose between future capabilities that improve the efficiency of the mission and the safety of their squad, and are encouraged to experiment with nonstandard tactics enabled by those capabilities.

HIVELET supports a range of different selection mechanisms (drawn from economic game theory) that impose different limitations on what capabilities players can bring on each mission. These methods provide guarantees that rational participants will honestly convey their priorities and preferences in the course of optimizing their own scores. Different selection mechanisms (such as auctions, alternating draft picks, and cake-cutting fair division methods) can be useful for collecting different types of data. In this application, the players used a random market—i.e., before each mission, the players are presented with a list of all available capabilities, each of which has been assigned a random price, as shown in Figure 4. They may select any number of those capabilities, but the prices are deducted from their upcoming mission score. In this manner, players are pressured to make do with as few upgrades as possible, driving them to think critically about the relative values of different capabilities. The random market method was used because it is quickly understood by novices and suitable for a single-player experience.

The capabilities available included RF sensors that help locate the objective, IR sensors that help identify potential hostiles, image processors that help differentiate civilians from hostiles, various control mechanisms for the personal drone, user interface displays available to display sensor data, and advanced munitions to give the players improved firepower. Players could combine these capabilities to support a range of strategies, both conventional and unconventional. For example, players might buy a "follow-me" control mechanism, an IR sensor, and an augmented-reality helmet display, then perform the mission on foot with a visual indicator of nearby potential threats (such a strategy is depicted in use in Figure 5). Alternatively, they could buy an onboard camera for their UAV, robotic underarms, and an onboard RF sensor, then attempt to find the objective and complete the mission entirely with the drone, without putting their own characters at risk.



RF Sensor

IR Sensor and Boosts

IR Sensor 20
None

IR Sensor +30 40
IR Sensor

IR Sensor +60 20
IR Sensor

PoseDetection 40
IR Sensor

PoseDetection Enchancement 10
IR Sensor
PoseDetection

IR False Negative Reduction 80
IR Sensor

IR False Positive Reduction 40
IR Sensor

Display Enchancements

Player On Map 80
None

Augmented Reality 5
Drone Camera

Drone Model

Quieter Drone 5
None

Algorithms improve the pose detection software, resulting in near-perfect friend-or-foe differentiation.

Points:
0

User ID
Enter Name

FIGURE 4. The technology selection screen is used by players to choose what capabilities they will combine for the next mission. Each choice is a capability (e.g., IR sensor) or an upgrade (e.g., +30 meter range to a sensor). The number to the right is an abstract resource cost that forces players to think critically about what capabilities are worthwhile.

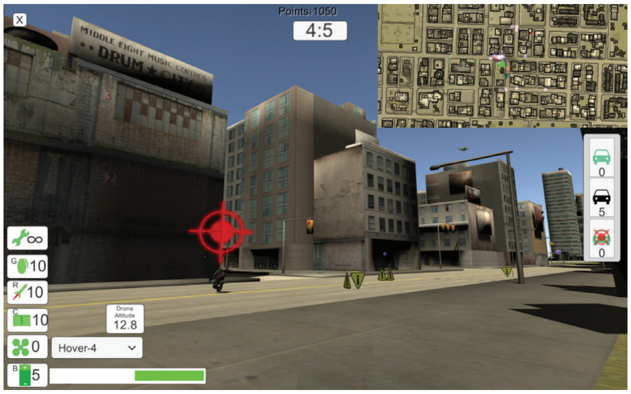


FIGURE 5. After selecting capabilities, players try them out in a real-time simulation of an infantry mission. In the depicted scenario, the player is clearing a route of hostile forces and buried threats with the help of a UAV-mounted sensor package. The player has to select a capability package that will support a balance between detection, confirmation, and response.

Bringing Quantitative Analysis to Early Concept Analysis

Much of the work thus far on HIVELET has been on validating its merit rather than on applying its technique to particular domains. Data collected from initial experiments indicate that the technique is capable of quickly providing useful quantitative data about the value of future technologies. In this section, we review some of the quantitative analyses that are enabled by this style of rapid-play serious game.

We assessed the utility of rapid-play serious games by looking at data collected from users who are interacting with the system, including participants with a mix of research and military backgrounds. From the data collected about player choices, performance, and behaviors, we can see that the technique is able to bring data analytics to bear on answering questions about future technology. Figure 6 shows that a few hours of gameplay is sufficient for players to start providing coherent data to be analyzed: 1 hour of training plus 1 hour of solo play was enough for players to stabilize their scores and start producing consistent levels of performance. Players' scores were calculated from a combination of completing the mission, avoiding enemy fire, and minimizing the number of technologies purchased. Players self-reported that 1 to 2 hours of exposure was sufficient to learn the game, formulate a strategy, execute the strategy, and develop opinions about the value of the technologies, at least within the context of the mission simulated in the game.

Once we believe that players have had sufficient time to develop opinions, we can examine what values they expressed. Figure 7 shows the frequency with which each of the 29 modeled technologies was selected across all participants, and we can see strong trends in player preferences within this mission context—finding a crashed airborne asset in a hostile urban environment. Drone-mounted cameras and long-range drone-mounted radio-frequency sensors were highly valued because they allowed players to quickly and safely scout for the lost asset. Interestingly, short-range drone-mounted RF sensors were considered to be almost useless, which helps us to establish the minimum acceptable requirements for such a device.

Drone-mounted IR sensors of any range were selected very rarely by players. This result initially surprised the research team as the IR sensors allowed

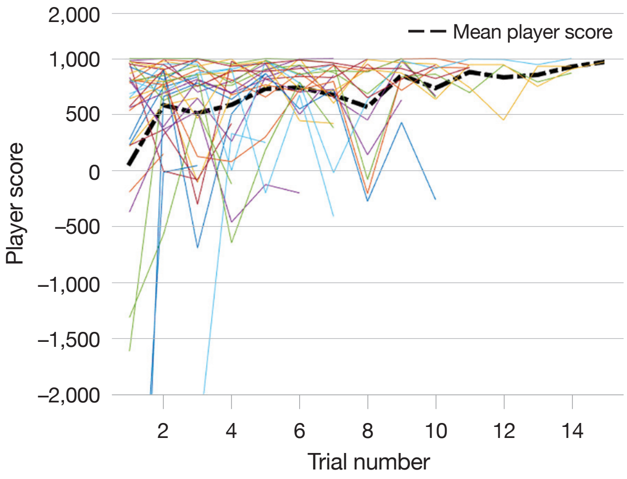


FIGURE 6. Novice players' scores improved and converged over the course of a 1-hour play session following a 1-hour training session. The maximum score is 1000, and the minimum is unbounded. Participants completed between 3 and 15 iterations within the allotted time. Those players who completed 10 or more iterations showed convergence, and those that completed fewer appeared to follow that trend. Qualitative surveys support the theory that a short session was sufficient for participants to formulate an opinion about how to incorporate the capabilities into a strategy and how much resulting utility those capabilities provided.

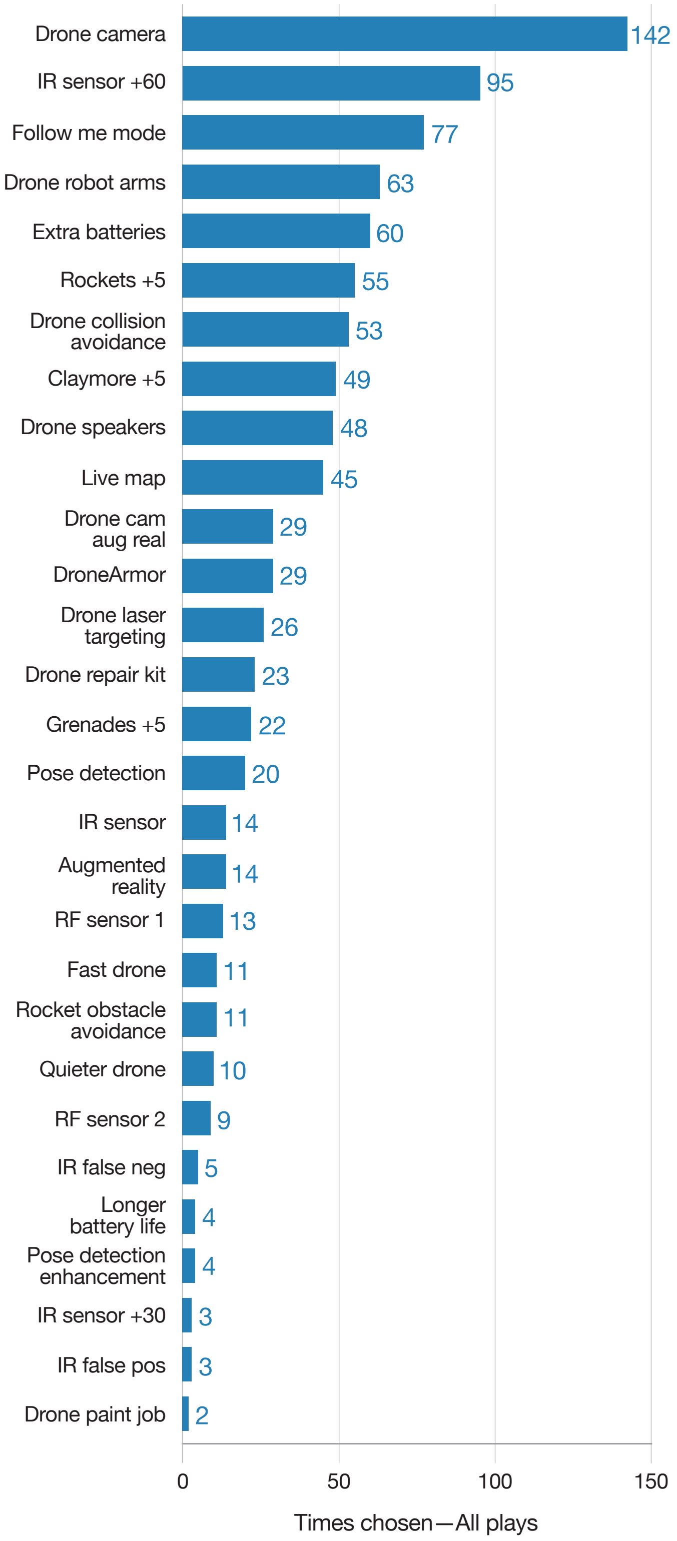


FIGURE 7. By logging the technologies that players selected, the prices that they were willing to pay, and the combinations that they often brought together, we can get a data-driven picture of the relative utility of the proposed technologies for the modeled mission. Some items are standalone capabilities while others represent upgrades to other capabilities, such as a higher quality (Grenades +5) or a higher range (IR Sensor +30). Players had access to a full description of performance characteristics.

players to know where hostile forces were in the city. This valuation makes more sense when paired with the qualitative feedback from players, who described the best strategy as running the entire mission with the personal drone and avoiding ever entering the city on foot. Thus, knowing the location of hostile forces was not important to this mission given the available technologies, and players discovered a strategy not anticipated by the research team. One of the strengths of rapid-play games is their ability to allow players to experiment with new strategies and anticipate how future technology will change tactics and doctrine.

Assessing players' preferences only makes sense if one believes that players are making good choices for themselves. To allay that concern, we can look for correlations in the data between players' preferences and their performance; such a correlation is shown in Figure 8. Even though the correlation is weak because of a limited data collection, the relationship in the data helps to validate two important assumptions: (1) in-game scoring motivates players to succeed, and (2) players are honestly expressing their opinions in the technology selection mechanism. We verified the first assumption by demonstrating that players change their level of risk aversion when the score penalty for coming under enemy fire is adjusted. Even with no real-world prize at stake, players who were given higher penalties for being shot within the game showed greater risk aversion in their behaviors and technology selections. We validated the second assumption by using technology selection mechanisms drawn from economic and mathematical game theory. We used methods that are known to encourage players to be honest in their assessments of value and to not incentivize gaming the system or lowballing a bid.

At this point, we have reason to believe that players are forming opinions in the time provided, that those opinions reflect actual utility within the game, and that

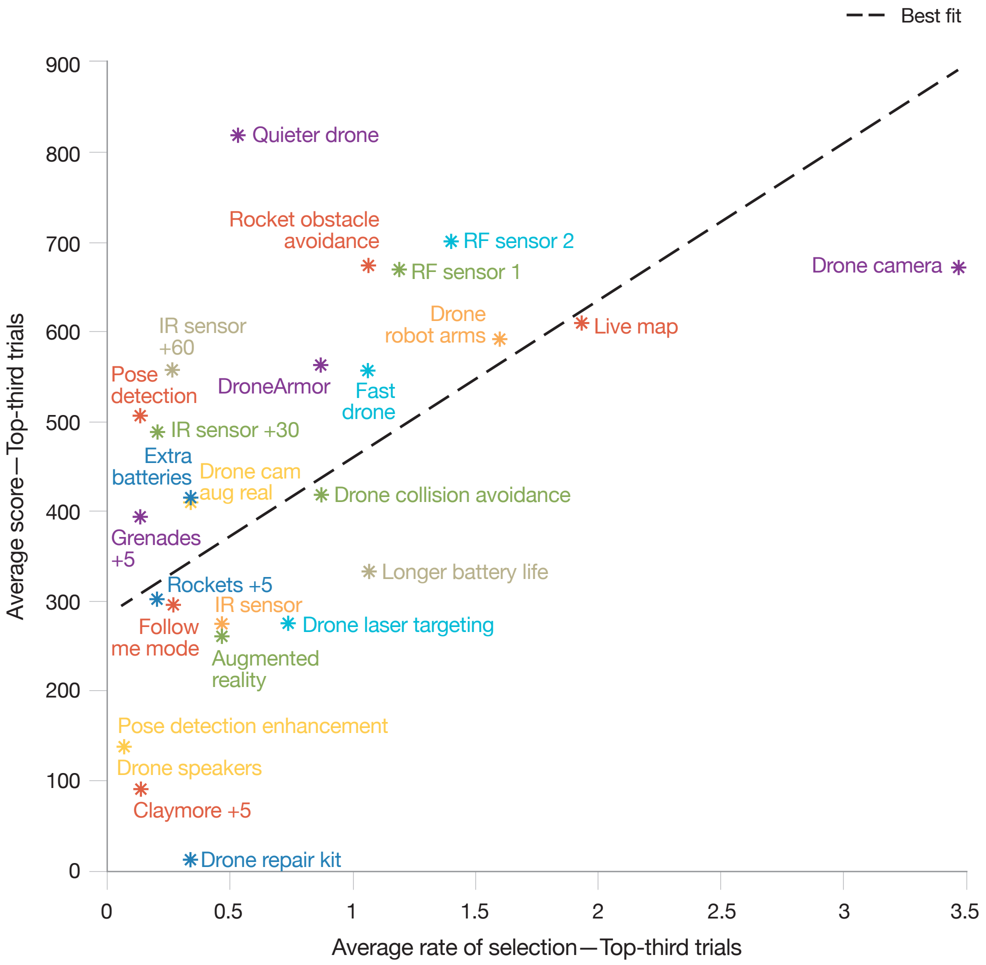


FIGURE 8. We see a correlation between the technologies players preferred to select and the technologies that produced better in-game performance scores. The vertical axis shows the average score when the capability was selected (max 1000). The horizontal axis shows the average number of times players selected the capability over all of their plays.

the game reflects realistic levels of risk aversion. So, we can trust the assessments players made of the modeled technologies, at least within the bounds of the mission they performed, the quality level used to model the technologies, and the correct calibration of the scoring incentives. As seen earlier, the strategies discovered by players sometimes surprise the research team, meaning that the method is capable of providing novel insights into how the technology will alter current practice.

Assessing the individual value of technologies is one thing, but part of the challenge of early-phase RDTE is looking at effective technology suites, that is, combinations of technologies or capabilities that will enhance performance. So, what we'd really like to discover is which technologies are synergistic, providing more value than the sum of their parts when deployed in concert. Figure 9 shows how data collected from rapid-play games might be used to answer that question by providing correlations in the selection of certain pairs of capabilities. In the illustrated example, there is a correlation between the use of drone-mounted cameras and drone-mounted manipulative arms, indicating that each of those technologies is more valuable when paired with the other. In contrast, technologies such as IR and RF sensors show no correlation—the value of each of those sensors is independent of whether or not the other is available.

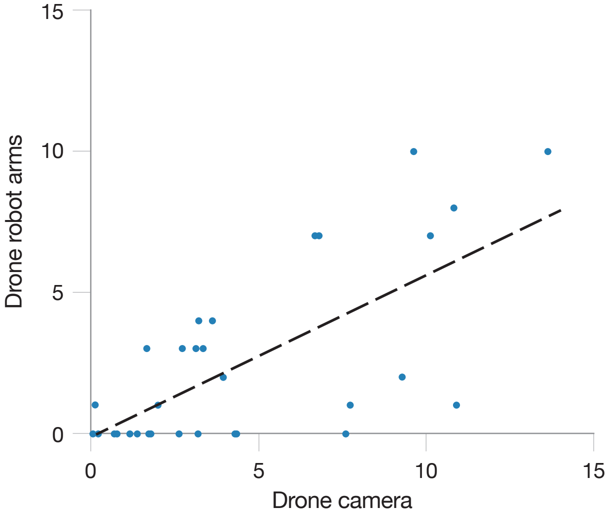


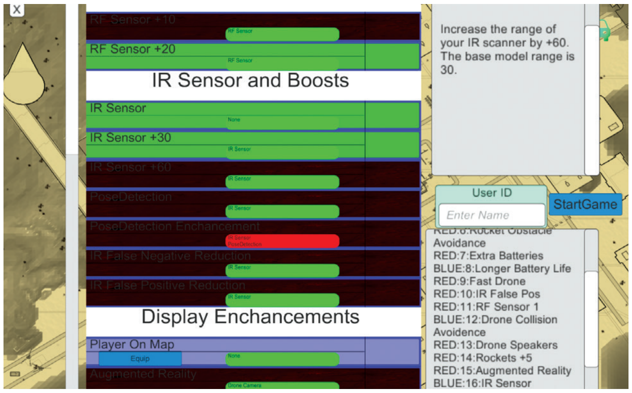
FIGURE 9. In this figure, each point represents a person. The *x*-axis shows how frequently the drone camera was chosen, and the *y*-axis is the frequency of selecting drone robot arms. The graph shows a correlation of preference for robot arms and preference for cameras, suggesting that the two capabilities are synergistic. These results are statistically weaker than the individual capability assessments because of the sample size used, but they indicate a promising possibility for what we can learn from data collected from rapid-play games.

Moving Forward

The broad field of serious games is growing but still early in its maturity. By and large, it has been established that digital games can be an effective tool for training users and changing their behavior, but techniques for doing so consistently and reliably are still an ongoing area of research [5]. The HIVELET work ongoing at Lincoln Laboratory aims to address that gap by providing and validating a framework for systematically modeling a domain and collecting useful data from it. In general, Lincoln Laboratory's work on serious games focused on making games a data-driven field for supporting quantitative analysis, thereby leveraging the Laboratory's data-analysis and domain-analysis strengths. Our view tends to be that a game is a sensor for measuring human decision making, thereby providing a quantitative way to study and learn from human experts. Thinking of a game as a sensor helps frame how it can be applied to systematically evaluating both technology and user performance.

Much of the research on serious games focuses on education, training, and medical therapy, and deals with the question of transference, that is, whether or not skills or behaviors learned in a game will transfer to the real world. A smaller portion of the field, including much of the research ongoing at Lincoln Laboratory, is examining the use of games in broader roles, such as domain analysis, technology evaluation, or crowdsourcing. Traditional tabletop games and professional wargames do explore all of those areas [6], but they are typically not executed in a data-driven or iterative fashion. Our continuing research effort is to tackle problems traditionally targeted by qualitative methods and supplement them with quantitative assessment from rapid-play digital games.

The HIVELET work done thus far has used a resource-constrained market as the selection mechanism that forces players to make cost-benefit assessments of proposed capabilities. A market method drives players to find a minimalistic solution that will let them succeed at the mission. Other selection methods drawn from game theory may be effective at collecting different types of data. For example, cake-cutting (where one player divides the set of capabilities into two groups and the other selects a preferred group) or drafting (illustrated in Figure 10) focuses players on what combinations of technologies are most synergistic or most redundant, and a draft (where players alternately select the available capabilities) focuses players on selecting flexible capabilities and building robust strategies that do not rely on any one capability being present. For different programmatic objectives, different techniques can be swapped into the framework to produce different types of data.



RF Sensor +10 — RF Sensor
RF Sensor +20 — RF Sensor
IR Sensor — None
IR Sensor +30 — IR Sensor
IR Sensor +60 — IR Sensor
PoseDetection — IR Sensor
PoseDetection Enchancement — IR Sensor PoseDetection
IR False Negative Reduction — IR Sensor
IR False Positive Reduction — IR Sensor
Player On Map — None
Augmented Reality — Drone Camera

Increase the range of your IR scanner by +60. The base model range is 30.

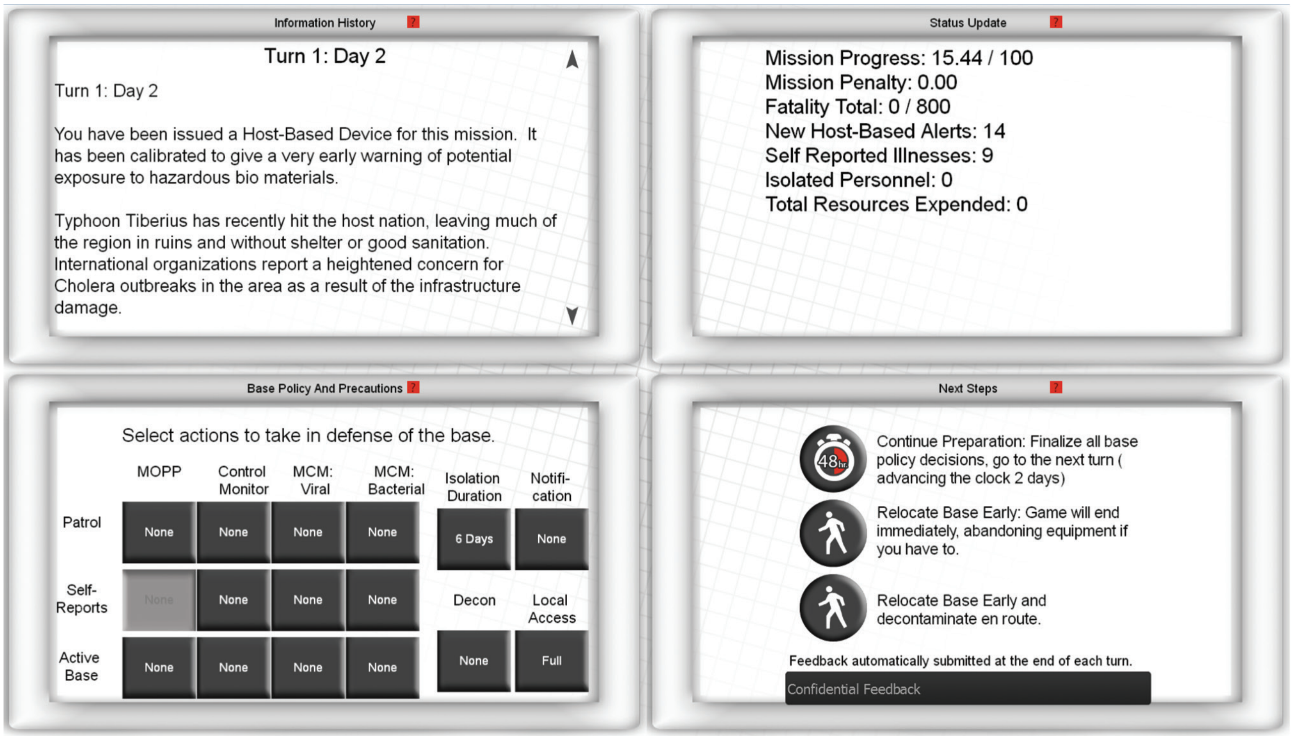
User ID: Enter Name

RED:6:Rocket Obstacle Avoidance
RED:7:Extra Batteries
BLUE:8:Longer Battery Life
RED:9:Fast Drone
RED:10:IR False Pos
RED:11:RF Sensor 1
BLUE:12:Drone Collision Avoidence
RED:13:Drone Speakers
RED:14:Rockets +5
RED:15:Augmented Reality
BLUE:16:IR Sensor

FIGURE 10. An alternate selection method drawn from game theory is a draft. There is no cost to selecting a technology, but each time the player takes a technology, the red force (either another human playing the adversary or a computer simulating an adversary) excludes three items from the list, forcing the player to prioritize selections and avoid brittle combinations.

The mission simulator described in this article was a 3D real-time model of tactical situations. The HIVELET approach can also be paired with turn-based strategic simulators that are used to assess how capabilities might impact higher-level decision making. Lincoln Laboratory has done prior work on rapid-play games for strategic-level decision making, such as the one shown in Figure 11. We have not yet combined such games with the HIVELET approach; analysis of the viability of such a combination is expected in the future.

The infantry example described earlier in this article focused on a single-player experience facing an automated threat. Multiplayer cooperative and competitive modes need to be explored further to determine if the HIVELET technique can also provide insight into how technology changes team dynamics and adversarial situations. Multiplayer implementation of HIVELET is not a technically challenging extension, but it complicates the



Turn 1: Day 2

Turn 1: Day 2

You have been issued a Host-Based Device for this mission. It has been calibrated to give a very early warning of potential exposure to hazardous bio materials.

Typhoon Tiberius has recently hit the host nation, leaving much of the region in ruins and without shelter or good sanitation. International organizations report a heightened concern for Cholera outbreaks in the area as a result of the infrastructure damage.

Mission Progress: 15.44 / 100
Mission Penalty: 0.00
Fatality Total: 0 / 800
New Host-Based Alerts: 14
Self Reported Illnesses: 9
Isolated Personnel: 0
Total Resources Expended: 0

Select actions to take in defense of the base.

	MOPP	Control Monitor	MCM: Viral	MCM: Bacterial	Isolation Duration	Notification
Patrol	None	None	None	None	6 Days	None
Self-Reports	None	None	None	None	Decon	Local Access
Active Base	None	None	None	None	None	Full

Continue Preparation: Finalize all base policy decisions, go to the next turn (advancing the clock 2 days)

Relocate Base Early: Game will end immediately, abandoning equipment if you have to.

Relocate Base Early and decontaminate en route.

Feedback automatically submitted at the end of each turn.

Confidential Feedback

FIGURE 11. This dashboard style interface is for a rapid-play game that focuses on strategic-level decision making. In this game, players are managing a forward-operating supply base that has potentially come under biological attack. Players use proposed future capabilities to help determine what precautions are appropriate and how much to jeopardize the mission to protect base personnel.

collection of data and thus may require many more plays before statistically meaningful conclusions can be reached. Research into the proper design of both the games and experiments will be important to broadening the work in that direction. Many emerging technologies focus on how multiple users interact, so providing quantitative support for the prioritization of technology that improves team coordination and effectiveness will be a growing field of interest that HIVELET aims to strengthen [7].

The most important piece of future work will be the application of the HIVELET technique to additional problem domains to refine and further validate the technique so that it can be integrated more smoothly into the RDTE process.

Acknowledgments

We would like to acknowledge the contributions of the HIVELET research team—Andrew Uhmeyer, Joel Kurucar, Daniel Hannon, Joseph Isaac, and Paul DiPastina; the HIVELET supervising committee—Timothy Dasey, Thomas Reynolds, Catherine Cabrera, Paula Ward, and Martine Kalke; and Kathryn Lannin and Dorothy Ryan, who provided technical writing input. ■

References

1. G. Klein, *Sources of Power: How People Make Decisions*. Cambridge, Mass.: MIT Press, 1998.
2. P. Suarez, "Games for a New Climate: Experiencing the Complexities of Future Risks," The Frederick S. Pardee Center for the Study of the Longer-Range Future, Task Force Report, Boston: Boston University, Nov. 2012.
3. P. Perla, *Peter Perla's The Art of Wargaming: A Guide for Professionals and Hobbyists*, J. Curry, ed. Annapolis, Md.: U.S. Naval Institute Press, 2012.
4. A.K. Dixit and B.J. Nalebuff, *The Art of Strategy: A Game Theorist's Guide to Success in Business and Life*. New York: W.W. Norton & Co., 2011.
5. T.M. Connolly, E.A. Boyle, E. MacArthur, T. Hainey, and J.M. Boyle, "A Systematic Literature Review of Empirical Evidence on Computer Games and Serious Games," *Computers & Education*, vol. 59, no. 2, 2012, pp. 661–686.
6. D. DellaVolpe, R. Babb, N. Miller, and G. Muir, *War Gamers' Handbook: A Guide for Professional War Gamers*, S. Burns, ed. Newport, R.I.: U.S. Naval War College, 2013.
7. S.C. Sutherland, C. Harteveld, G. Smith, J. Schwartz, and C. Talgar, "Exploring Digital Games as a Research and Educational Platform for Replicating Experiments," *Proceedings of the 2015 Northeast Decision Sciences Conference*, 2015.

About the Author

Robert M. Seater is a researcher in the Informatics and Decision Support Group at Lincoln Laboratory. He currently works on serious games, requirements analysis, and software engineering. He has applied serious games to a range of topics of interest to the Department of Defense and Department of Homeland Security, including the integration of unmanned aerial vehicles into infantry squads, large-scale emergency response, chemical and biological defense, and naval missile defense. He holds a bachelor's degree in mathematics and computer science from Haverford College and a doctorate from MIT in computer science and requirements engineering.

Serious Games for Collaborative Dark Network Discovery

Matthew P. Daggett, Daniel J. Hannon, Michael B. Hurley, and John O. Nwagbaraocha

Illicit social networks, such as trafficking or terrorist organizations, are difficult to discover because their clandestine nature limits their observability to data collection. Technological advances in remote sensing and analytical software can reduce the time- and human-intensive nature of network data curation and analysis, if effective human-system integration is achieved. To better understand this integration, researchers at Lincoln Laboratory created a succession of serious games to investigate methodologies for developing user-centered tools and quantitative human-system instrumentation, with the goal of improving network discovery. These games were employed in a multiyear study of team analytical performance and collaborative decision making, encompassing more than 80 teams and upwards of 400 unique players.

»

For decades, governments, militaries, researchers, and other organizations have focused significant resources toward the collection and analysis of information about illicit human social networks, such as gangs, cartels, traffickers, and terrorists. These networks, often referred to as dark networks, are difficult to study because their clandestine nature limits their observability to various data collection means and often precludes a full accounting of the network membership, structure, function, and dynamics [1–3]. Historically, the social sciences have provided the foundation for the study of dark networks, largely through the time- and human-intensive manual collection and curation of qualitative network data. However, this approach is not efficient, does not scale to large organizational studies, and generally only represents static points in time [4–6].

Over the past two decades as asymmetric conflicts and complex humanitarian crises have become more prevalent across the world, increased emphasis has been aimed at characterizing dark networks that operate in urban settings to perpetrate acts of violence, such as vehicular-borne explosive attacks, i.e., car bombings. The use of vehicles to facilitate explosive-laden attacks goes back to the 1920s and has been responsible for asymmetric attacks ranging from the Provisional Irish Republican Army's bombings during the Troubles in Northern Ireland in the 1960s to widespread explosive events by terrorist organizations during the conflicts in Iraq and Afghanistan in the last 15 years [7]. When a car bombing occurs, it can be extremely challenging for law enforcement to piece together information to determine which vehicles, facilities, and people were involved in the attack (Figure 1). This challenge is compounded by urban settings that allow perpetrators to flee and meld back into the background populous. In the last decade, advances in airborne remote sensing and terrestrial surveillance have made it possible for military and police agencies to observe not only the execution of these types of attacks on urban areas but often the events and coordination directly before and after.

However, the ability to triage surveillance video and imagery along with other reporting data—such as news, law enforcement reports, or social media— immediately after an attack is laborious and often requires teams of individuals to sift through large amounts of data to discover pieces of relevant evidentiary information [8]. Additionally, the discovered information must then be deconflicted, analyzed, validated, and synthesized to make timely risk-informed decisions about potential follow-on courses of action. It is unclear how and in what ways these teams should organize and operate, and what roles analytic and decision support technology should play in making these operations more efficient and effective.

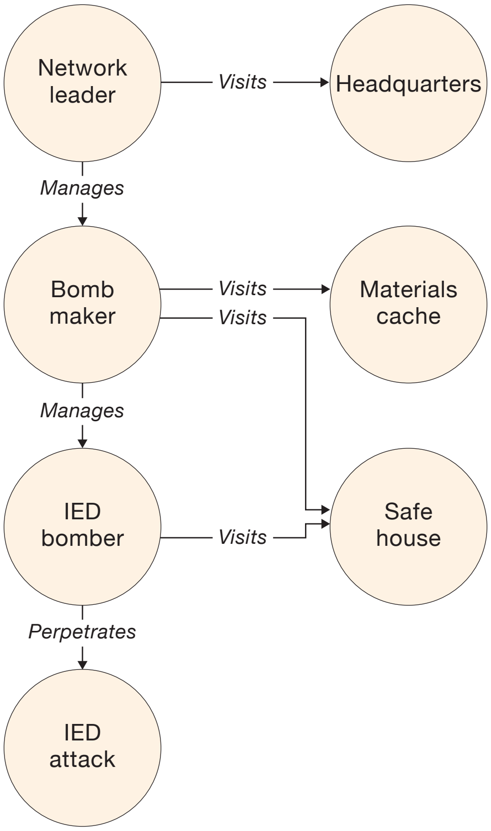


FIGURE 1. In this example of a small vehicular-borne explosives graph network, the circles (nodes) represent people, locations, and events, and the lines (edges) that connect them correspond to the nature of the association between the nodes. Arrows on the lines represent the directionality of the relationship.

Game Design

In 2009, we and other researchers at MIT Lincoln Laboratory developed a serious game to address some of the challenges regarding clandestine network discovery. We created a platform to better understand how a team of players uses multimodal geospatial data to discover information about a dark network and synthesizes those data to make decisions [9–11].

Serious Games for Research and Development

Since the 1950s, Lincoln Laboratory has performed applied research and development for national security missions on a foundation of rigorous systems analysis, full system prototyping, and development of long-term advanced technologies. As the discrete systems of earlier decades have been replaced with complex interconnected systems of systems, traditional modeling and simulation and systems analysis can be insufficient because these methods often fail to properly account for human dynamics. To overcome these limitations, researchers at the Laboratory developed a suite of methodologies and technologies to design serious games that can be used as tools to model, experiment with, and assess complex human-system dynamics that approximate those of realistic sociotechnical enterprises. In serious games, gameplay is used to achieve an explicit purpose other than amusement. We have used such games across a spectrum of the research and development process, including experiential learning, concept exploration, requirements analysis, tool development and evaluation, human performance assessment, and decision analysis.

Research Objectives

We identified four research objectives for this serious gaming work:

- Games as analysis tools. We wanted to demonstrate the value of using serious gameplay as a systems analysis tool for human-intensive workflows and applications.
- System requirements derived from decisions. In remote sensing research and development, the process often

starts with an understanding of the phenomenology of the sensing environment and observables of interest. This phase leads to the development of sensor hardware that is then integrated and fielded on the premise that the sensor capabilities are inherently useful; however, many sensor systems have not been jointly developed alongside the decision processes their data are meant to inform. In this work, we wanted to essentially invert this development and acquisition process by starting with an understanding of what information is needed to make decisions and work backward to build an end-to-end workflow that results in actionable information. Then, we could use the gaming process and simulation capabilities to determine what the technical and performance requirements should be for both the sensors and their data analysis systems.

- Effective game development scope. We wanted to learn how to build the right level of realism and fidelity into the game to create enough immersion and engagement to force players into an effective decision process, while limiting the scope and cost of development.
- Rapid tool and workflow utility assessment. Through the use of robust human-system instrumentation to collect quantitative human performance data, we wanted to develop an end-to-end process to assess the value and utility of tools early in their development cycle.

Scenario Development

During the design phase of the game, we spent a lot of effort on generating the requirements for the storyboard (hereafter referred to as the scenario) that drives the game data generation and game mechanics toward achieving the research goals. The most important requirement of the scenario design involved four elements of the geographic location of the game:

1. The game should be based in an area of future strategic importance to the U.S. government. At the time of design, many activities within the Middle East were within the purview of the U.S. Central Command, and we decided to focus instead on Africa because the U.S. Africa Command had just been established in October 2008.
2. The location should be in an area within Africa that is unfamiliar to most players, including potential players with a good understanding of global geopolitics or with prior military experience. This condition minimizes the effect of experiential knowledge and preconceptions about the scenario.
3. The area should have a history of instability and violence to build the scenario around, as well as a complex environment of actors composed of the indigenous population, foreign fighters from neighboring areas, an external coalition military presence, and multiple nongovernmental organizations (NGOs).
4. The scenario should be focused on a city with a compact, dense urban core that quickly fans out into a suburban and then rural expanse. This constraint limits the scale of the geographic area of regard for the game participants and aligns with the field of views of the sensor concepts to be used in the data simulation.

On the basis of these criteria, we chose a moderately large city in a landlocked country in Central Africa (hereafter referred to as the city). When this scenario was developed in 2009, the city had a recent history of instability. It had been briefly seized by insurgents in 2006, and in 2007 local rebels had declared war on foreigners and refugees from the surrounding region, requiring the deployment of thousands of international peacekeeping troops in 2008.

In the city, several prominent groups formed what we called the red, blue, gray, and white actors; this color naming convention is derived from military wargaming nomenclature. The red actors are those operating to incite violence in the city, such as the local rebel group, who is seen as anti-government and anti-foreigner and who has staged many recent attacks through car bombings and kidnappings. The blue actors work to counter red groups and include an international coalition of peacekeeping forces headquartered in the city and the game participants themselves. The gray actors are those who have an unclear affiliation with a side, such as the national army, who is undisciplined and believed to be heavily infiltrated by rebel groups. Lastly, the white actors consist of various NGOs and news media in the region.

From research into the city's historical events and groups, we constructed a timeline that laid out a sequence of activities that would take place in the scenario. Next, data from a geographic information system were analyzed to determine both public locations, such as the city's airport or the local army garrison, and private locations, such as previous weapons caches used

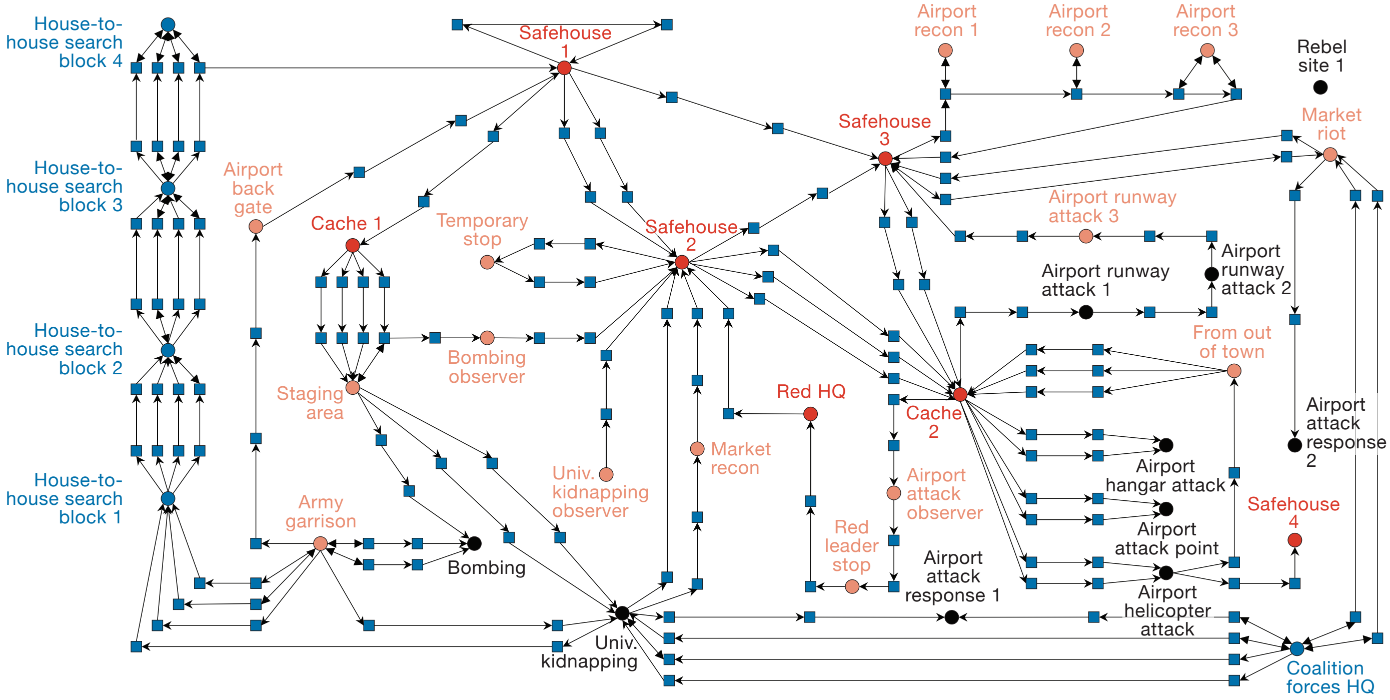


FIGURE 2. The graph of the scenario shows the network of facilities and the vehicle journeys, or tracks, that visit them. The circles (nodes) represent locations visited by a vehicle, such as a safe house or weapons cache, and the squares represent an intermediate stop of a vehicle. The lines (edges) that connect the circles and squares correspond to discrete vehicle tracks between two locations, and arrows represent the directionality of the tracks moving between the sites.

by the local rebel group and places that could be sites of interest within the scenario. Care was taken to make sure the locations chosen for these red actor sites had no known associations with public locations in any of the information sources examined.

With the locations of interest chosen, we scripted a series of activities that formed the scaffolding of the scenario, which broke down into three waves of activities. The first wave started with a truck bombing followed by a kidnapping at the local university. Next, the kidnapping prompted a neighborhood search by the national army and the discovery of a red safe house, necessitating movements of multiple red actors to other locations. In the final wave, certain groups staged a riot at the main city market to divert attention away from a coordinated attack on the airport that included the bombing of the runway and a nearby hangar. Next, we designed an intricate series of timed vehicle journeys, or tracks, between all of these event locations and other locations, such as staging areas or headquarter compounds; these tracks formed the basis of the networks of vehicles and facilities associated with the red actors. To add complexity to the scenario we gave many of these vehicle journeys intermediary stops and starts or circuitous routes between clandestine facilities, as these diversions are typical operational security principles. The final scenario consisted of nine hours of activity and comprised 37 networked sites, 27 of which were associated with the red network. Seven of the sites were high-value red facilities, eight sites were associated with clandestine red activities, seven were associated with overt red activities, and five were innocuous red vehicle stops. A graphical depiction of the network of these events, sites, and intermediary stops, which became the basis of the scenario truth, is shown in Figure 2. For simplicity, the figure does not show the times associated with each of the movement starts and stops. Hereafter, the terms scenario vehicles or scenario sites refer to those associated with the red network and not those of the background actors or their activity.

Remote Sensing Concepts

Starting in the mid-2000s, large-format airborne imaging systems were being developed and fielded for military and other applications. These systems used multiple large-format optical focal planes to capture oblique panchromatic imagery of the ground from an airborne platform in a circular orbit. Through sophisticated orthorectification algorithms and supercomputer-class processing hardware, the systems stitched all the raw data into large mosaiced images that appear as if they were collected from directly overhead. These early systems, which could produce imagery at approximately 0.5 meters per pixel at about two frames per second over small city-size fields of view, were termed wide-area motion imagery (WAMI) sensors [12]. While WAMI sensors were an amazing achievement in optical engineering and image processing, it was unclear at the time how best to make use of these nascent capabilities and the large volumes of data they produced.

When designing this game, we wanted to explicitly explore the applications of WAMI to the problems of network discovery and so made motion imagery the primary mechanism by which data about the scenario network were gathered and provided to players. We chose a sensing concept in which WAMI is collected from a hypothetical sensor over an area of interest that is 5 kilometers by 5 kilometers, which would have the majority of the roughly 8-kilometer-by-8-kilometer urban core of the city continuously within the field of view. A graphical depiction of this area is shown in Figure 3.

Several hypothetical collection concepts of operations were explored, including the real-time downlink of small chips of imagery that are a subset of the full sensor field of view and the traditional paradigm of offline data processing and the use of WAMI in a forensic capacity only. We also developed a companion sensing concept for an airborne ground moving target indicator radar that would provide coverage of up to a 20-kilometer-by-20-kilometer field of view in the suburban and rural areas surrounding the city. However, in early testing of this concept, users struggled to interpret and make sense of this nonliteral data modality, and it was later removed from the game to focus on the higher priority task of determining the best utility for WAMI.

Game Implementation

Data Generation

With the scenario and remote sensing concepts developed, we produced datasets that would become the primary sources of information used during gameplay, specifically a set of vehicle tracks, a multiresolution corpus of motion imagery, and a series of alert messages to cue teams to activities within the data.

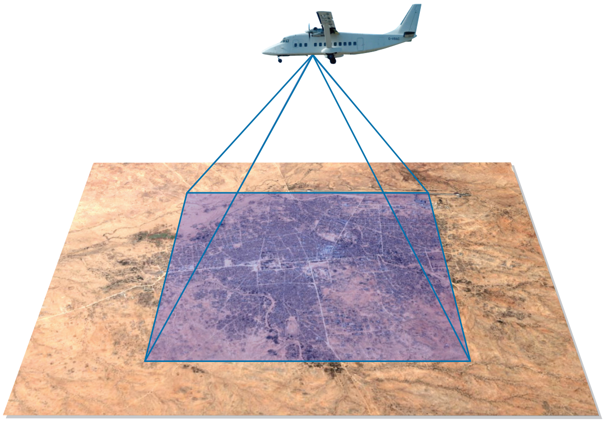


FIGURE 3. The illustration shows the sensing concept used in the game. The projected base image shows the urban core of the city and the superimposed blue box represents the instantaneous field of view of the wide-area motion imaging sensor on board the aircraft. The aircraft orbits around the perimeter of the city while the urban core remains persistently within the field of view.

The first step in the data generation process was to obtain a multispectral satellite image of the city from a commercial vendor to use as the basis for all other data products. The image was used to assign physical locations to the sites and events from the scenario, in congruence with the appearance of those locations within the imagery; for example, safe houses were chosen at locations of remote walled compounds. Next, we used geographic information system tools to develop a road network by tracing out all the primary, secondary, and tertiary roads.

With these data, we generated a vehicle track dataset by using a commercial vehicle-motion modeling and simulation tool that uses a road network, waypoints, and vehicle-motion models to generate track data through time. The scenario timeline and geographic locations were used to construct waypoints for the vehicles associated with the scenario activities, and the waypoints evolved through multiple runs of the modeling tool to match the scenario to the physics of the vehicle-motion simulator.

Next, background vehicle tracks representing the gray and white actors were embedded with the scenario tracks to create a realistic and more complex traffic environment. To create the background activities, we developed a statistical model to estimate a rough distribution of residences and workplaces across the city. Vehicles were modeled as starting from randomly selected residential locations at a distributed set of times in the morning of the scenario with a destination randomly selected from the workplace distribution. A series of pauses and additional waypoints were then randomly selected for each vehicle to complete its waypoint list for the game duration. If a vehicle completed its waypoint list before the end of the scenario, it repeated the list until the scenario was over. This list of tracks and waypoints was then run through the same vehicle-motion modeling tool as used for the scenario tracks. To avoid confusing the game players and incurring possible false team decisions, the start, stop, and waypoint locations for background vehicles were filtered to reject areas that were at or near any of the static red scenario locations. Lastly, the track data were run through a process to apply noise to the true vehicle dynamics and to break tracks into multiple segments, thus mimicking the problems associated with optical multitarget tracking systems of the era.

Next, to generate the motion imagery dataset, we leveraged a technique from early video game graphics by which two-dimensional bitmap images, or sprites, are embedded into a larger image and then rendered as a single scene. To produce the base image, a graphic artist modified the original satellite image of the city to erase any vehicles visible on roadways and adjacent to sites associated with vehicle tracks in the scenario. Additionally, any people visible were also removed because the sensing concept used in the game instructs users that people are not visible at the resolution used. Next, exemplar vehicles, such as cars and trucks, were extracted from the unmodified satellite image and turned into sprites. To produce the simulated vehicle movements, the vehicle track positions at each time step in the scenario were turned into pixel locations in the modified base image, and the vehicle sprites were rotated to the direction of travel and inserted onto the base image. The resulting new image was rendered with vehicles embedded. This process was repeated for each time step of the scenario to generate a full-scale motion imagery dataset.

Lastly, we developed a dataset of text reports, or messages, to help give context to activities in the motion imagery data and to help keep the teams focused on the game objectives since teams will frequently get stuck on red herrings with the sensor data alone. In conjunction with the scenario creation, messages were written to tip the players to events of interest in the imagery, such as reporting of overt attacks. Each message contained information about the originating source—for example, regional news organizations, local law enforcement, NGOs, and coalition military forces—and about the time and location, with varying degrees of precision, that the text referenced. Some events generated multiple messages from multiple sources, requiring players to assess each message's relevance and veracity with respect to the objectives of the game.

Game Architecture

Because we wanted to employ a large degree of video data manipulation and collaborative tools and interfaces, we were unable to find an existing game development framework that met all the requirements, so we developed our own purpose-built game architecture. The approach was to push as much of the processing and display tasks to server-side components so that the game client could be made lightweight and responsive to players. Additionally, we wanted all game state information stored on the server so that if players accidentally closed their game client, it could restart right where players left off with no information loss (this feature is especially important in teamwork settings).

A game client named Bluestreak was developed in Java and built around NASA's WorldWind, an open-source software development kit for visualizing and hosting geospatial data in a 3D globe-like interface [13]. A description of the user-interface features and a screenshot are discussed in Figure 4. In addition to the individual client features described in the figure caption, another major capability was the ability to collaborate across Bluestreak clients; for example, when a user made a placemark, i.e., a geospatial bookmark, on one client, that object showed up on all other clients, greatly improving shared awareness that underpins effective collaboration. Also included was a set of interfaces that the teams could use to codify their final decisions to enable automated scoring of their answers. All user actions executed in the tool, such as user-interface state changes, and all polling events, such as the geospatial and temporal extents of the current data displayed in the map, were recorded with specialized software instrumentation.

The game server consisted of three major components: a specialized imagery and geospatial data server, a relational database, and a web service communication

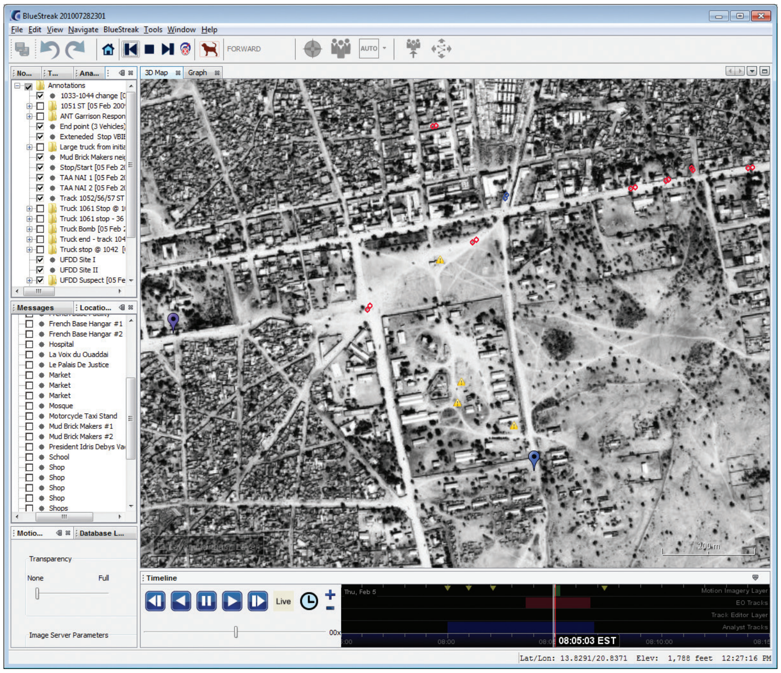


FIGURE 4. In the Bluestreak game client, the center map display fills the majority of the user interface and is flanked by configurable panels on the left and a custom timeline control on the bottom. The user-interface panels on the left are user configurable to enable viewing of additional layers of data on the map display, including data provided as part of the game and data generated by players. Provided data include geographic information system data, such as named areas and locations of interest relevant to the scenario, or text displays that show reports received as part of the scenario. User data can be geospatial polylines of vehicle movements, called tracks; geospatial bookmarks made by users, called placemarks; and other information. The timeline control allows users to manipulate the rendering of imagery, vehicle tracks, and other data by using a single temporal extent or selectable time range. This function, which gives users the ability to scrub forward and backward in time and see patterns in the data as they render on the screen, is especially useful for analyzing the movement behaviors of vehicles.

channel. The generated WAMI data was passed to Bluestreak through a custom-built JPEG2000 image server, designed to scale to multiple streams of imagery data sent to tens of clients. Hereafter, the term *video* will be used to refer to the viewing of these streams of motion imagery. The base satellite imagery and other geospatial data were served via an open-source web mapping system called MapServer [14] and translated into a pyramid of multiscale image tiles that can be efficiently passed to all the game clients for display. All game geospatial, message, and instrumentation data were read from or recorded to a PostgreSQL relational database, with PostGIS spatial database extensions. Lastly, publish and subscribe web service interfaces were used to transfer the data between the game server and game clients.

Human-System Instrumentation

From network operations control centers to expeditionary military detachments, teams of humans interoperate with complicated systems to create complex sociotechnical enterprises. Within these enterprises, the most critical component of overall performance is that of the humans, yet their contribution is often the least understood. Traditional measurement methodologies, such as human observation, are often subjective and anecdotal and can suffer from biases and differences in interpretation. Additionally, existing tools to measure human behavior can be qualitative and are insufficient in capturing intricate dynamics within an individual (intra-individual) and between individuals (inter-individual). Lastly, the time- and human-intensive collection of these data does not scale to large organizational studies. These limitations hinder the ability of researchers to draw objective conclusions and understand the parameters influencing team success.

Over several years, we have developed a data-driven research methodology and technical framework, Humatics, to address the aforementioned challenges by quantitatively measuring human behavior, rigorously assessing human analytical and cognitive performance, and providing data-driven ways to improve the effectiveness of individuals and teams. Humatics incorporates three major areas of research: system-level, physiological, and cognitive instrumentation; assessment methodology and metrics development; and performance feedback and behavioral recommendation. Figure 5 depicts our instantiation of this approach and its application to the study of teams' abilities to effectively discover data, make sense of those data, and make decisions in the context of a serious game.

The development of an instrumentation and data collection strategy for a given human-system research effort requires a careful consideration of the specific learning objective for the process under study and the identification of observables to be measured to enable insight. A measurement strategy can then be based on which method and phenomenology are best suited to directly or indirectly measure those observables. For our research, specific instrumentation modalities were chosen to augment qualitative human observations with nearly continuous collection to enable the analysis of dynamic low-level behavioral signals.

The first element of the framework in Figure 5 is the instrumented analyst workstation, where both system-level and physiological instrumentation are used to characterize human-system interactions. System-level instrumentation is accomplished through the insertion or enabling of software code that logs graphical user interface interaction events, queries to and transactions with databases, the data visible to the user, and more. To add context to the data, screen recordings are continuously captured and a research-grade eye tracker detects the user's location of gaze on the screen. This physiological information is used for cross-referencing the system-level data.

The next element is cognitive instrumentation, which is used to measure behaviors associated with the cognitive processing of information. To quantify the comprehension and situational understanding of teams during scenario-based training or serious games, knowledge elicitation techniques are employed [15, 16]. Measuring a player's or team's understanding requires explicit elicitation of information from individuals through a series of free-response and targeted multiple-choice or Likert-scale questions that are focused on the concepts of comprehension and inference development. Comprehension is a measurement of the facts presented in the data (e.g., who, what, when, and where), such as the location and time of an attack, and an inference is a measure of a player's interpretation of the data (e.g., how and why), such as who a player believes facilitated the attack and the attacker's possible motive. In addition to its use for gaze tracking on the screen, the eye tracker is used to perform pupillometry (precise measurement of the pupil's diameter) to noninvasively estimate human cognitive load [17], another facet of cognitive instrumentation.

The last framework instrumentation modality uses wearable sensors called sociometric badges [18] to record nonlinguistic metadata of speech behaviors, body movement, and other data. Originally developed by the MIT Media Laboratory, the badges have often been employed in longitudinal studies of the communication patterns of large organizations. We used badges with modified firmware and custom post-processing software

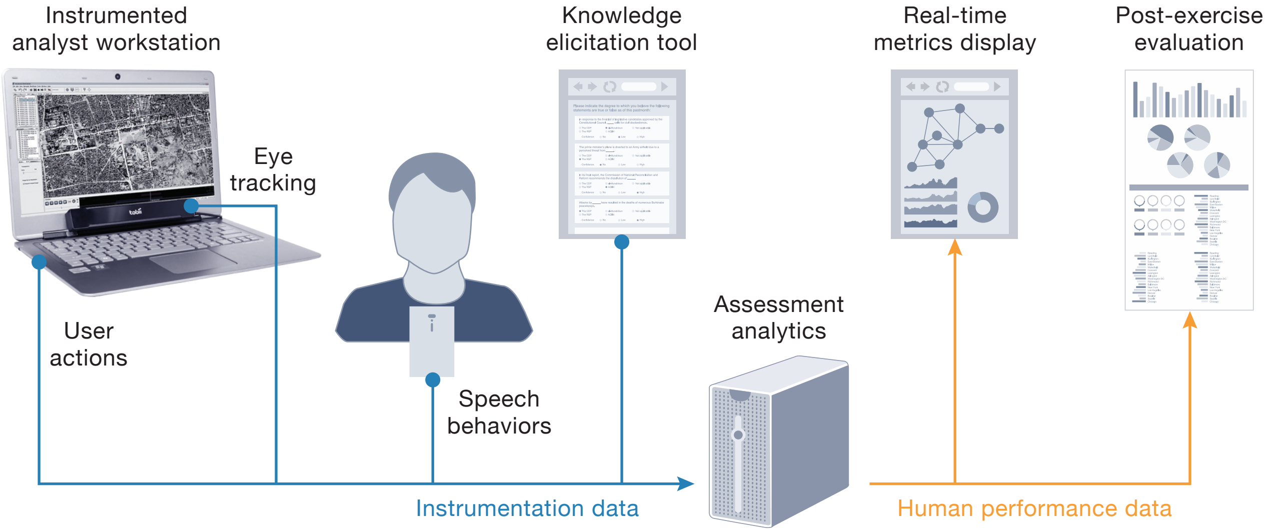


FIGURE 5. This diagram depicts the Humatics framework—a platform to measure and make sense of human analytical performance data. System-level, physiological, and cognitive sensors and instrumentation are used to produce rich quantitative data of human-human and human-system interactions. Instrumentation data are jointly processed with advanced metrics and turned into measures of human performance that are visualized in custom displays to provide performance feedback and pinpoint areas for behavioral recommendation.

to increase granularity for small group dynamics within hierarchical teams.

Our collected instrumentation data were processed with specialized metrics and used for real-time diagnostic displays or post-experiment assessment. Real-time displays allow for immediate team evaluation to enable behavioral redirection, while offline post-processing supports in-depth analysis and process improvement. Our team assessments are an example of the latter.

Mechanics and Gameplay

White Team

The role of the white team (not to be confused with the white scenario actors) is to ensure the smooth, effective operation of the game. This oversight includes monitoring the physical setup and tear down of the gaming facilities, preparing and presenting all materials, conducting briefings and training, and generally facilitating the overall event. As facilitators, the white team answered questions about tool use and reminded teams about overall objectives, but they did not give away information about the scenario or provide feedback during gameplay about the relative effectiveness of different strategies. The white team provided real-time assessment at the end of the game and briefings of the results to the different team members. White team members included many of the original game developers and other staff who have extensive experience with the game.

Game Event Timeline

After the initial test versions of the game were employed, we honed in on a game event process that would allow for four to 12 competitive teams per day to play through the exercise, depending on available hardware infrastructure and white team members, with a game event lasting one or two days. More than 80 teams and 400 participants have played this game over the life cycle of this research effort, and we have analyzed in detail a large subset of these teams.

We began the game event process by obtaining informed consent from the participants in accordance with approved human-subject research protocols. During different phases of this research, we recruited subjects from a wide population that included college students, scientists, engineers, professional military analysts, military instructors, and senior government officials. Next, the participants received introductory briefings that highlighted the research purpose and goals, and provided background knowledge, such as a primer on social network analysis. The network analysis primer is critical to success in the exercise because it introduces concepts about how people and facilities are associated in a network, what differences exist between static and transient location types, how leadership is often isolated within dark networks, and how to build and interpret graph network diagrams. After the background presentations, participants received a live plenary tool demonstration, followed by a mission briefing that oriented them to the scenario and tasks they would be required to perform. This presentation was designed with the look and feel of a military-style mission briefing, with fragmentary operational orders defining the rules of engagement, an overview of the city and its destabilization, an overview of the remote sensing and other data capabilities available to teams, and a review of possible end-state courses of action and recommended decision criteria.

Next, individuals were assigned to teams, known as the blue teams, through a process that used limited demographic data collected during orientation to attempt to balance the team members' backgrounds, skill sets, seniority, and organizational affiliations. Teams then moved to separate rooms where they could play the game and deliberate in private, and where individualized training on the game tools would take place. The white team used training checklists to ensure that each participant had a minimum proficiency with the game software. Next, a team strategy session took place, and teams prepared for a practice scenario. The purpose of the practice scenario was three-fold: to try out the plans of action that teams developed in the strategy session, to identify any areas of training that needed reviewing, and to be familiarized with each facet of the gameplay. A second team strategy session allowed teams to discuss what went well and what went wrong during practice, and regroup before the start of the main exercise.

During a short break after the main exercise, the blue teams moved back to the plenary room while the white team scored and analyzed the teams' performance. In a following "hotwash," a representative from each team explained to all participants what that person's team believed happened in the scenario and what approach that team took. Then, the white team gave the scenario reveal, which described step by step all the activities within the scenario and the information that teams should have found and the decisions that they should have made. Finally, the scoring and performance assessment results were presented and winners received an inexpensive trophy. In practice, we found that teams compete fiercely for the chance to win even an inexpensive trophy, and organizational affiliation and pride also significantly affect team competitiveness and engagement.

The usual game block lasted four hours, with the main exercise taking up about one and a half hours; generally, two game blocks were performed per day with as many as six concurrent teams per block. Some of the earliest games required eight hours of gameplay per team, but we later switched to a shorter, simplified game format to focus on specific teamwork and decision-making facets of the game and to yield more games played per game event.

Blue Team Strategy

One of the challenges of collaborative games is that team members often do not know one another or have not worked together previously. Because this arrangement can lead to ineffective team dynamics, one of the purposes of the two team strategy sessions is to force a dialog between the individual players to get them to think about team structure and roles. During these sessions, we instructed the teams to consider these five major facets:

- Approach. Teams should think about the scenario briefing and decide on an initial concept of operations, which they can later refine in the second strategy session once they've tried it in practice. Members should also discuss their assumptions about the scenario, their risk tolerance, and other factors so that there is less potential for conflicting ideals later in the game. They should also decide if they want to use some of the automated tools provided in the game software or stick with a more manual tradecraft.
- Resource allocation. Teams are provided one less game workstation than the number of team members, so they need to decide how to allocate their human and compute resources. In early testing, we found that if we gave every player a workstation, the members failed to organize into a team, and by having one less game client than players, hierarchies formed with one player taking a leadership and integration role and the rest taking on the discovery tasks. Teams also have the option of not using all of the workstations, and some opt for a pair programming model with two players collaborating around a single workstation.
- Team roles. Teams need to decide who plays what roles, generally leader and worker roles. Leaders solicit workers for the latest information to synthesize into higher-level meaning and also often serve a scribe function by categorizing this information on the provided whiteboard or other means. The worker roles break out into a multitude of possible tasks, such as tracking vehicles from source to destination, watching for new messages to alert the team, and building the network, either on a whiteboard or in the graph tool in the game client. Multiple players may take on any of the leadership or worker positions, and it's up to the team to self-organize their gameplay.
- Collaboration. Teams must ask how they will function and collaborate on the tasks that need to be performed. For example, who will assign tasks and track their status, and who will monitor work that has been done? Because the game client provides a number of ways to annotate with text and color the information discovered and input it into the software, teams should discuss naming and color conventions, such as putting player initials on information or using the color of annotation to label potential decision criteria.
- Decision making. Teams must decide how to select a course of action related to the sites they have discovered in the game. They should discuss if they want to make decisions as they discover new information or wait until the end to take stock of all available information. They should also determine how aggressively or conservatively to play, judging how their decisions and the ensuing risks and rewards impact scoring and game performance.

Gameplay

After all the training, practice, and strategy sessions, gameplay on the main scenario began with two to six concurrent competitive teams. Teams contained between three and eight players, with the standard configuration being five—four players on computers and one team leader. The task given to the teams during the mission briefing was to uncover as many of the sites (locations) used by the red network to perpetrate the attacks in the scenario, and then to make recommendations on a course of action for each discovered location by the end of gameplay. Within the game were two main phases: the discovery phase, in which players analyzed the video, track, and message data to discover red activities and their associated locations, and the decision phase, in which players synthesized their collected information and adjudicated their uncertainty and risk to choose courses of action. How teams moved between the two phases was one of preference: some teams spent the first 80 percent of gameplay discovering information and the last 20 percent making decisions, and other teams assigned potential courses of actions to sites as they discovered them and continually adjudicated those decisions throughout gameplay. A visual depiction of the game workflow, broken down by the two phases, is shown in Figure 6.

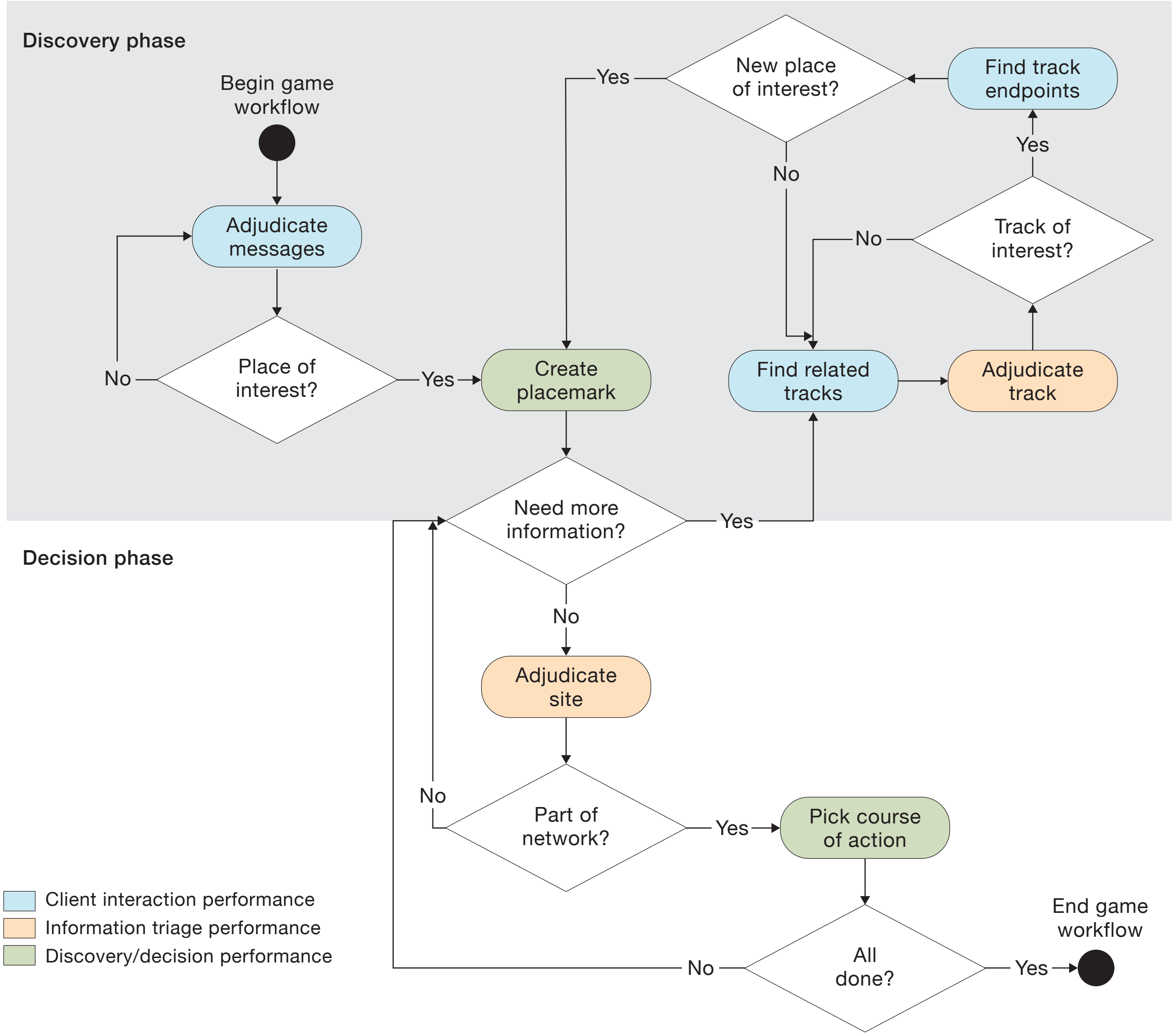


FIGURE 6. A canonical workflow diagram of gameplay provides a visual of which steps in the workflow map to specific measures of human performance in the game. The top half of the diagram shows the discovery phase of the game, during which players triage and make sense of game data to discover the network of actors and facilities they are trying to uncover. The lower half of diagram represents the decision phase of the game, during which players adjudicate the information they discovered and make risk-informed decisions regarding which locations they believe are part of the scenario network and how strong a course of action should be taken against those locations. Each of the three different types (colors) of game performance was the focus of a large human-subject experiment and assessment.

Photo: U.S. Navy (a)

Photo: U.S. Navy (b)

FIGURE 7. Participants engaged in an exercise with the Naval Special Warfare Command. The analysis discovery phase of the exercise is picured in (a), and (b) shows the later decision-making phase.

As the scenario began, teams were alerted in real-time to events unfolding in the scenario via messages that arrived and were cued to the place and time in the video associated with the messages. Players observed the events in the video and adjudicated the relevance and veracity of the associated messages since messages can be factual or ambiguous depending on the message source. If the location of the activity in the message was of interest, players made a placemark there and then queried for tracks that either originated or arrived from that location. They then determined if any of the tracks were associated with the event through spatio-temporal analysis of the video. Players followed tracks associated with the previous red activity to their source or destination and entered placemarks at those locations to indicate potential association with red activity. As the scenario evolved, more messages came in, cueing players to other locations of both relevant and nonrelevant activities. Through the association of video vehicle tracks with their user placemarks, players built out the network of red sites. Teams could catalog their understanding of the network as it evolved by using tools within Bluestreak or on the provided whiteboards and large-format notepads.

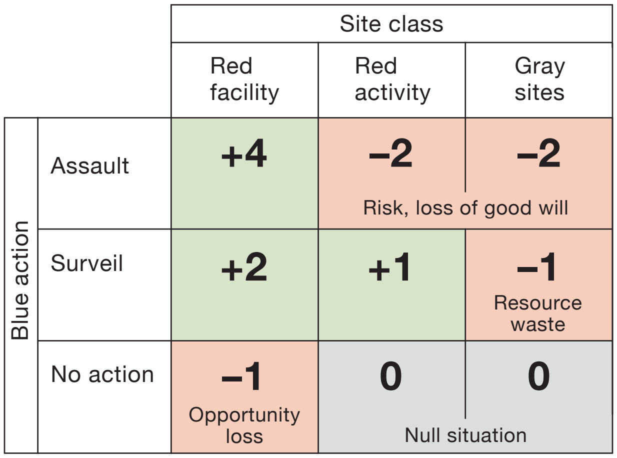
As the teams entered the final decision phase, they went through each of the placemarks believed to be associated with the red network, discussed the evidence they had accrued about that site and the courses of action they should take, and then chose from three potential actions in the placemark menu:

- Assault. Sites that should be assaulted are those that have a static association with the red network and that, if law enforcement or military were sent to interdict these facilities, would certainly reveal red personnel or material. Examples of sites to assault are safe houses, weapons caches, and the red headquarters.
- Surveil. Sites for which the team cannot determine if they should be assaulted or regarded as transient sites associated with temporary red activity, and should be nominated for continued surveillance because they may be associated with red activities in the future. Examples of surveil sites are attack staging areas, the garrison of the local army, and long-duration stops by red vehicles.
- No action. No action should be chosen for all placemarks that are not associated with the red network and are innocuous.

This process continued until all placemarks were adjudicated and courses of action chosen, with no action being the default action. As teams approached the expiration of game time, team dynamics became very animated, often with heated discussion and a frenetic pace of locking in and checking all course-of-action choices. An example of gameplay can be seen in Figure 7.

Team Scoring and Evaluation

Depending on the game event, teams are evaluated across multiple performance factors, including decision making, information discovery, and verbal communication, with team decision performance as the primary mechanism for declaring a winner. After the teams



		Site class		
		Red facility	Red activity	Gray sites
Blue action	Assault	+4	−2 Risk, loss of good will	−2
	Surveil	+2	+1	−1 Resource waste
	No action	−1 Opportunity loss	0 Null situation	0

FIGURE 8. In the scoring matrix used to adjudicate the decision-making performance of teams during the game, the columns represent three classes of locations, or sites, and the rows correspond to three levels of courses of action the teams can assign to each instance and class of sites discovered during the game. Cells shaded in green indicate that the team's chosen course of action was appropriate for the respective class of site, resulting in a gain of points, and red cells represent a course of action that was inappropriate, resulting in the loss of points. Gray cells represent action and class mappings for which points were neither gained nor lost.

finished the game and their decisions were stored in Bluestreak, a server-side scoring script was run to take into account several factors, such as geospatial closeness, to determine which teams correctly identified the location and value of each of the red sites in the scenario. A scoring matrix was used to award points to each correctly identified site and points were subtracted for incorrect decisions, as detailed in Figure 8.

For red facilities, the correct decision in the game was to assault, and it earned the most points. If the facility was surveilled instead, then half the point value was awarded because some information was gained, and if no action was chosen, then points were deducted because the opportunity for some discovery was lost. For red activities, the correct action was to surveil them and points were awarded accordingly. If a red activity was assaulted, points were deducted because this action added risk to the interdicting force and lost good will with the local population when an innocuous location was assaulted. If no action was chosen for red activities, then points were neither awarded nor deducted. For gray sites, or those involving the background populous, points were deducted for an incorrect assault and for a surveil because these actions lost good will and wasted surveillance resources. The correct action for all gray sites was no action.

The weights of the points between the levels of the courses of action and their correct and incorrect value were constructed to match the concept of the scenario while also prohibiting teams from trying to "game" the game. Point values for the red facilities and red activities were totaled into a single score for each team, and the team with the highest score of the game event won. Often, scores could be negative if teams were aggressive in their approach, and if a tie occurred, additional performance measures were used to break the tie.

Experimentation Phase 1: User-Centered Tool Development

Considering the work involved in the development of the sensor and traffic simulation models and the complex scenario, we knew that completing this game would be challenging and that some tooling and automation would be required, especially with respect to information organization and knowledge management, for the game to be effective. However, rather than building those capabilities into the initial iteration of the game software, we wanted to use this opportunity to learn new methods for designing effective human-system tools.

In general, users are ineffectual at explaining to others what is hard for them and what types of capabilities would improve their work process. Frameworks like user-centered design have gone a long way toward analyzing and envisioning how users are likely to use technology, and then validating those user behavior assumptions with real-world tests and evaluation [19]. In our case, because we were working with a new type of data, WAMI, with no established workflows and best practices, explicitly studying the intended user was not straightforward. Instead, we wanted to see if gameplay could be used to implicitly learn what tasks were hard for users and where in the process there was friction. Our approach was to study user solutions to the game in the absence of the needed tooling and then turn our observations and user artifacts into a requirements specification for developing new user-centered capabilities. Once those new capabilities were developed, we could use the same methods to deploy the capabilities, measure their utility, and retool them to be more effective.

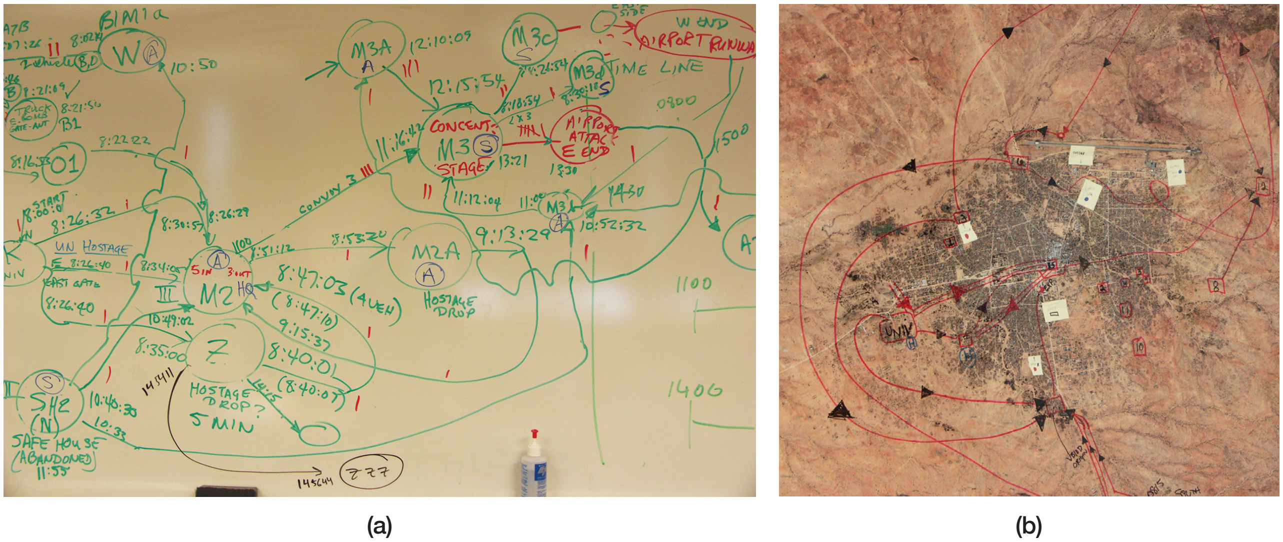


FIGURE 9. Teams used a whiteboard and map to manually organize information during gameplay. The image in (a) shows a node-and-link network diagram representing different sites (circles) discovered in the game data and the vehicle tracks between them (lines). Also annotated on this diagram are names given to each of the sites and tracks by the teams, and the start and stop times of the vehicle journeys. The image in (b) shows a geospatial network view of similar information using markers and sticky notes on a laminated map.

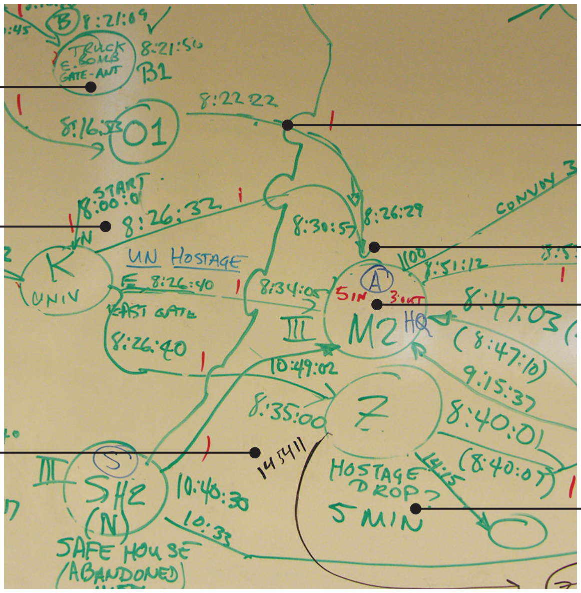
Experiment Design for Requirements Generation

In this phase of research, we wanted to better understand how users might best use real-time video information to aid network discovery during an unfolding event. We designed an experiment in which users had a static base satellite image on their map display and the ability to overlay streams of up to eight real-time 100-meter-by-100-meter video chips from the airborne WAMI sensor's field of view. Users could slave those chips to follow a specific vehicle or persistently stare at a location on the ground. In this construct, players not only had to manage their human and compute resources but also their sensor resources. While video was only available within the eight available chips, track positions of vehicles were available across the entire sensor field of view. However, when vehicles stopped within the scene, the track broke and started with a new track identifier as the vehicle started moving again, thus requiring teams to devise methods for how best to mentally stitch all these tracks back into a single vehicle journey. We knew bookkeeping was going to be a challenge in this experiment but wanted to see the methods that teams came up with through gameplay.

A series of team games was deployed, and we used both photographic and room video recordings to track how teams discussed and captured information via the whiteboard and hardcopy maps. Among the many different approaches to capturing and coding the game network information were the two examples of this instrumentation seen in Figure 9.

By studying how teams solved various problems through different methods on the whiteboard and paper map, we determined the requirements for a set of tools that users would have liked to have had during the exercise. Figure 10 shows how a team's map suggests ideas for a new tool. In this example, a player could benefit from a network visualization tool that is integrated with the map and track paradigms within the game client. The requirements for the tool fall into three groups of network information representation:

- Node information. Users would like the ability to customize the names of sites (nodes) with their own annotations and to represent track metadata, such as the duration of a vehicle stop, as attributes of a particular node.
- Link information. Users would like the ability to visualize track metadata along links (tracks) between nodes (sites); such metadata could include name annotations, departure and arrival times, autogenerated track identifiers associated with a track, and the number of track (vehicle) counts between two nodes.
- Graph layout. Users would like to represent



User annotations of sites and tracks

Departure and arrival times for vehicles

Bookkeeping of system track IDs

Minimization of link crossings in the layout

Directionality of tracks

Vehicle counts to and from nodes

Indications of vehicle stop durations

FIGURE 10. Studying how teams manually organized their information can provide insight on ways to improve information management through new tool development and optimization of existing capabilities. In this example, the callouts detail software requirements or features that would address some of the information management and visual layout needs of building a network diagram of sites of interest and the vehicle tracks that transit between them.

source-to-destination directionality of tracks and to minimize the crossing of links in the graph representation.

Development of Network Analysis Tools

Our requirements development process led to three new major features that were added into Bluestreak and the back-end game architecture:

- Joint space-time queries. In the initial iterations of the game, if players had interest in vehicles that may depart or arrive from a site of interest, they would have to scrub through all of the temporal extents of the data to find tracks. To increase the efficiency of this operation, we created a feature called Nomination to allow players to choose a point on the map, a temporal extent, such as 30 minutes before and 30 minutes after the current time step, and a geographic radius, such as 50 meters around the selected point. The game server would retrieve all tracks that matched that joint space-time query and display those to players.
- Track repair tool. As mentioned in the section on data generation, track breaks were introduced to mimic the real-world performance of optical multitarget tracking systems of the day. With those systems, tracks would manifest as single source-to-destination journeys and comprise multiple track segments, requiring players to monitor which track identifications corresponded to which vehicle journey. To improve this process, we developed a tool named Bloodhound to allow players to use the video data to positively identify when the same vehicle is responsible for the end of one track and the start of another. Bloodhound then lets players stitch those two system tracks into an analyst track, greatly simplifying the information management and network representation.
- Integrated network visualization tools. As shown in Figures 9 and 10, organizing and visualizing all of the information related to the sites and tracks that form the scenario network requires a lot of effort and bookkeeping to be useful for unraveling the game scenario. The new Nomination and Bloodhound features enabled the players to quickly find tracks associated with points of interest and quickly repair them from source to destination, allowing the network to be rapidly built out and effectively visualized. We developed two graph visualization tools, one to produce abstract node-and-link diagrams and one to produce a geospatial node-and-link diagram showing the spatial representation of the sites and tracks in network. An example of both representations can be

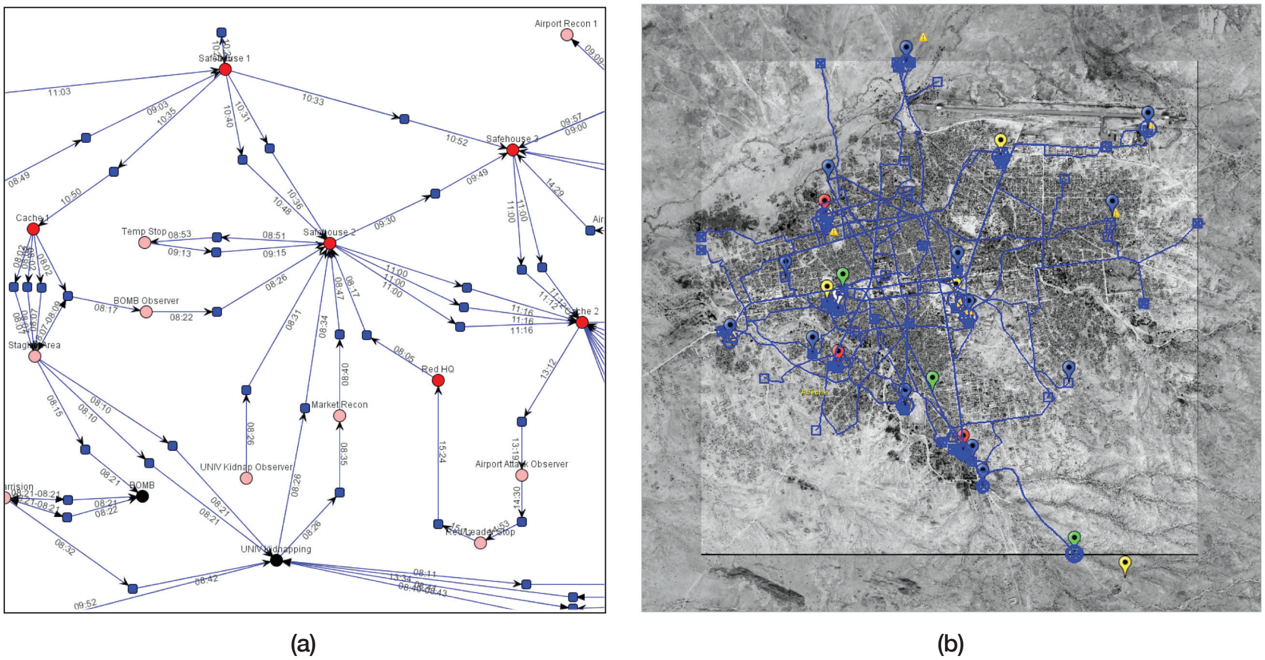


FIGURE 11. Abstract and geospatial graph representation tools are developed through a user-centered requirements process. In the abstract graph view (a), the circles (nodes) represent locations visited by a vehicle, and the squares represent an intermediate stop of a vehicle. The lines (edges) that connect the circles and squares correspond to discrete vehicle tracks between two locations, and arrows represent the directionality track moving between the sites. Similarly, (b) is the geospatial graph view. The blue circles, squares, and lines have the same connotations as the symbols in the abstract view; however, the edges now follow the full geospatial extent of the tracks they represent. Additionally, the multicolor pushpin icons represent placemarks of interest to the team.

seen in Figure 11. While abstract node-link diagrams have been used for a long time, the geospatial graph was entirely novel at the time of development. Lastly, one additional key feature of the abstract graph is that it was built to be fully collaborative with the other game clients, so when one player moved a node on a client, the node also moved on all the other game clients, allowing teams to have true shared representations.

Utility Assessment

After some initial user testing of the new tools, a series of game exercises was employed to assess the utility of these tools in improving the abilities of teams and reducing some of the human-intensive aspects of the network discovery and information management. During the debriefings from these exercises, we found that in general the players really liked the Nomination feature to find tracks associated with a site of interest. However, the judicious use of this feature had an unintended consequence. The tool developers thought that after a Nomination was executed and the results returned, it would be convenient for the player to have the site and tracks associated with the Nomination automatically placed on the graph. But this automation ended up cluttering the graph displays with both user-placed and system-placed information, with no clear distinction between the two. Once this clutter occurred, the team stopped using the graph tool and went back to using the whiteboard because that was a representation over which they had full control. One player described the automated placement function as similar to using the top of a desk to store documents that need to be read, without realizing that other people would constantly place other documents on the desktop, rendering it useless as an organizational mechanism.

To fix the placement problem, we added a step that asks users after they make a Nomination query if they also want the results added to the graph. The graph tools then started to provide great utility for network organization, and several winning teams in this testing phase used it exclusively. A comparison of a graph cluttered by the system and one built solely by players is depicted in Figure 12. This example of gameplay forcing users to

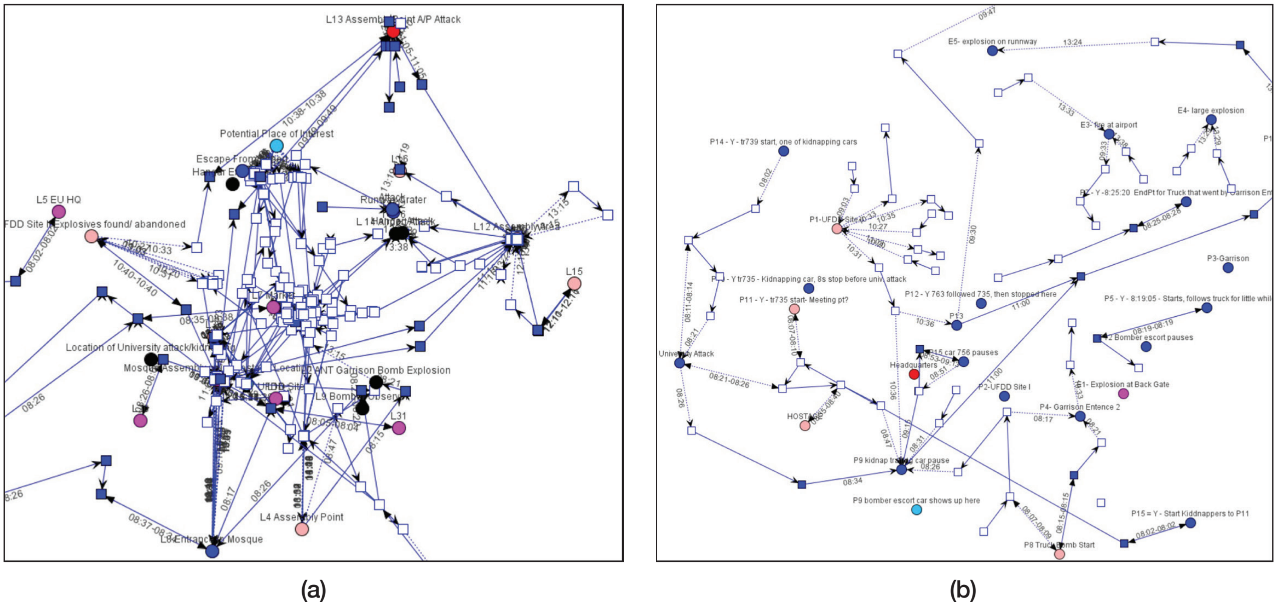

L13 Assembly Point A/P Attack
Potential Place of Interest
L5 EU HQ
UFDD Site Explosives found/ abandoned
L12 Assembly Area
L15
Location of University attack/kidnap
Mosque Assembly
Location NT Garrison Bomb Explosion
L9 Bomb Disposal
L31
L4 Assembly Point
L8 Entrance to Mosque
(a)
E5- explosion on runway
E4- large explosion
E3- fire at airport
P14 - Y - tr739 start, one of kidnapping cars
P1-UFDD Site 1
P3-Garrison
P12 - Y 763 followed 735, then stopped here
P5 - Y - 8:19:05 - Starts, follows truck for little while
P11 - Y - tr735 start- Meeting pt?
University Attack
Headquarters
P2-UFDD Site I
E1- Explosion at Back Gate
P4- Garrison Entence 2
HOSTAGE
P9 kidnap truck car pause
P9 bomber escort car shows up here
P15 = Y - Start Kiddnappers to P11
P8 Truck Bomb Start
(b)

FIGURE 12. Network graphs are generated by players during utility testing of new tools. The circles (nodes) represent locations visited by a vehicle, and the squares represent an intermediate stop of a vehicle. The lines (edges) that connect the circles and squares correspond to discrete vehicle tracks between two locations. The diagram in (a) is from a game in which automation added information to the user's graph, inadvertently cluttering the workspace with information of unknown provenance and limiting the utility of the tool. The diagram in (b) is a user graph from a subsequent exercise in which players were given the option to accept or reject automated information, leading to much more effective use of the tool because of a greater understanding and trust of the automation.

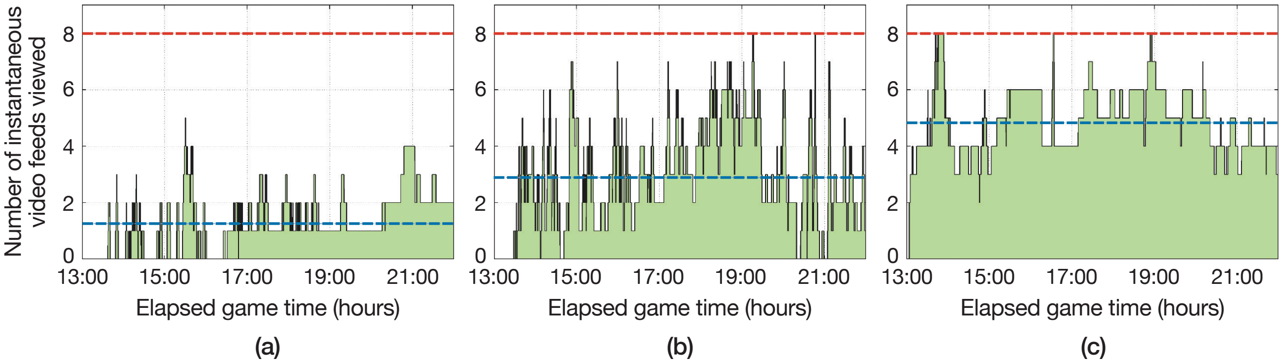


FIGURE 13. The plots provide an analysis of video feed utilization by game players. The *x*-axis represents the elapsed time during gameplay. The *y*-axis represents the number of instantaneous video feeds being viewed by a team at a given time step, with 0 representing no feeds in use and 8 representing all the feeds being used. The maximum number of feeds is represented by the dashed red line and the average number of feeds used is represented by the dashed blue line. Case (a) shows video utilization with the baseline tooling that precluded effective use of the video feeds, with an instantaneous average of 1.3 video chips. Case (b) shows improved video usage after new tools were deployed in a subsequent game to better integrate the video into the workflow and reduce the human-intensive nature of using the video, with an instantaneous average of 2.9 chips. Case (c) shows increased video usage after refinements were made to the new tools in response to targeted user feedback, with an instantaneous average of 4.8 chips.

evaluate what parts of new features have utility and which need refinement or reimagining was much more efficient and effective than simply asking users how they might make use of a particular tool or feature.

In addition to qualitative analysis of user experience, we leveraged the game client instrumentation to characterize how well players used the eight available video chips described in the experiment design. In the initial games without the improved tools, the imagery analysis and information management tasks dominated the teams' time, precluding their ability to make best use of the eight possible video feeds, as shown in Figure 13a.

The plot in Figure 13b details increased video utilization after the deployment of the new network analysis tools (Nomination and Bloodhound) that improved the integration of the video feeds into the workflow. The plot in Figure 13c shows both additional increased video usage after the user-feedback process led to the refinement of the new tools and more effective use of the graph to organize and prioritize work.

Besides analyzing video utilization, we looked at how well teams kept up with real-time information as the scenario evolved because this ability is often associated with stronger game performance and decision making. Instrumentation was built into the Bluestreak game client that logs both the elapsed game time when players are looking at data and the point in the scenario timeline that the data are referencing. An example of this instrumentation data is shown in Figure 14. As the scenario began, an abundance of activity caused teams to spend a lot of time looking forensically at older data to orient themselves before they felt comfortable reviewing new data arriving in near-real time. Through our analyses, we found that the addition of the three network analysis tools shortened the amount of game time teams spent observing data forensically before they transitioned to real-time operations.

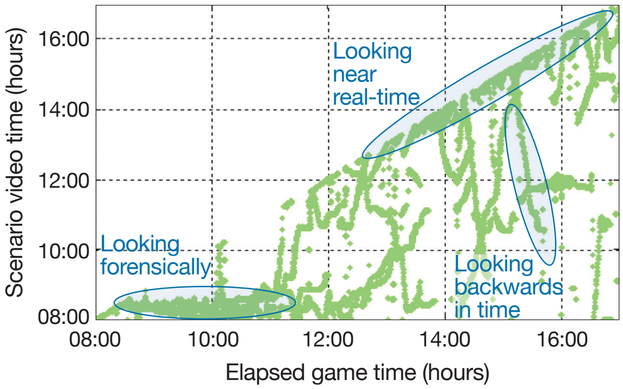


FIGURE 14. The plot depicts an analysis of how timely teams were at keeping up with real-time game data as the scenario evolved. The *x*-axis represents the elapsed game time from the beginning to the end. The *y*-axis represents the time in the scenario that the data references. Each green dot represents a record of these two timestamps, achieved through the software instrumentation within the game client. A team analyzing the data in real time would create green dots across the diagonal, and any dots below the diagonal represent a forensic examination of data. As teams oriented to the initial set of activities in the scenario, they began to view the arriving data and often did deep backward dives in time to assess all activities at a particular location.

Experimentation Phase 2: Assessing Teamwork and Decision-Making Performance

A major finding from our first phase of experiments was that team dynamics played a critical role in the outcome of the game, and anecdotally we could often predict just by observing the strategy sessions and gameplay which teams would do well at decision making. We had teams that were introverted and precise in their coordination and communication, and we had teams that were verbose and constantly challenging each other's assumptions; both of these team dynamics were found to be successful. The success of two almost opposite styles of teamwork made us want to understand on a granular level the underlying factors that influenced success. We designed a set of experiments to study teamwork and its effect on decision making. For these experiments, we modified the game format and employed the Humatics human-system instrumentation framework to augment our qualitative human observations with quantitative, persistent, and objective measurements of human-system behavior. By jointly processing the collected multimodal instrumentation data, we could make holistic characterizations of human-human and human-system interaction. The fidelity and granularity of these data were informative and, in some instances, could predict performance in the activities being measured [20].

Experiment Design for Instrumenting the Analysis and Decision Processes

To implement the second phase of experiments, we modified the format of the game to emphasize the team collaboration and decision-making components and to reduce human-intensive data analysis aspects of the original game. In this second format, we made the following primary modifications:

- Shortened the length of the scenario by half so that gameplay and the overall game event would be shorter.
- Changed the sensing concept to make all motion imagery available to players across the entire field of regard at the start of the game; having all the video data rather than only eight small time-based video chips would increase information discovery.
- Replaced the track dataset, including its track breaks and sensor ambiguities, with the ground truth track

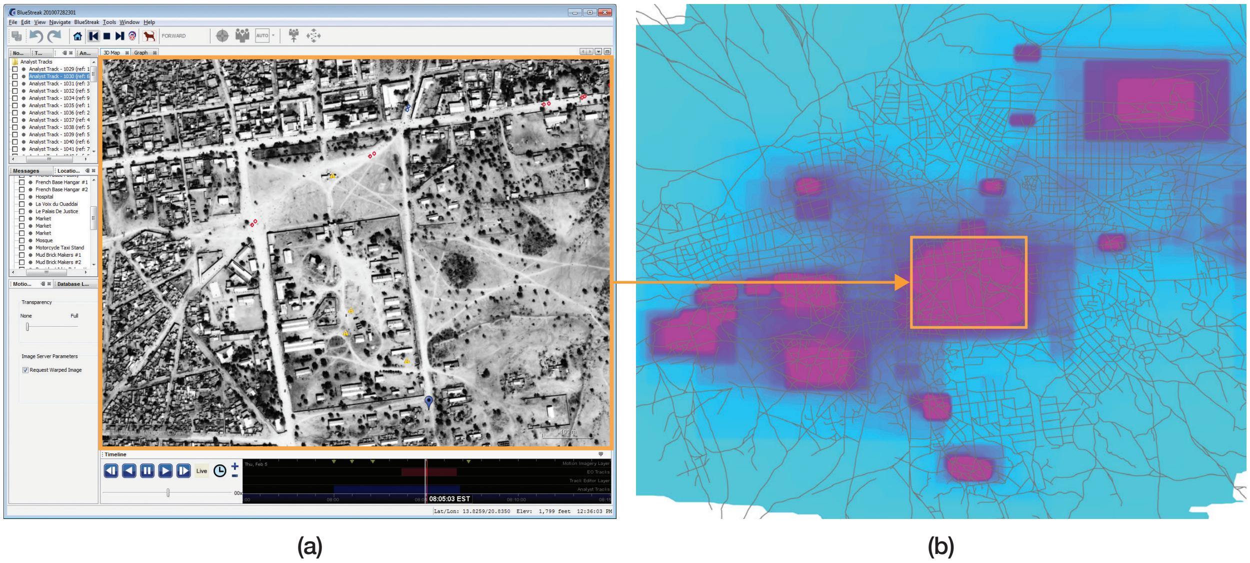


FIGURE 15. In this illustration of viewport instrumentation, the game client (a) is viewing a portion of the video data, whose viewport, or geospatial extents, are represented by the orange box on the game client and heat map (b), as indicated by the orange arrow. In the viewport heat map, areas of magenta represent areas in which the teams viewed a large amount of video data, whereas the cyan areas indicate areas looked at infrequently. These heat maps can be used to understand a team's geospatial analysis strategy.

data to allow players to focus on determining the connections between source and destination locations rather than spending a lot of time stitching together broken tracks.

- Increased the number and relevance of the game messages to ensure teams focused on the game objectives.

We also made improvements to the staging of the game event by standardizing training processes, materials given to teams, and types of information provided by game docents to players. The intent of this standardization was to reduce as much variability in the game events as possible so that the game could be run many times with different teams to produce a dataset for follow-on human-subject research.

Team Assessment Case Study

To assess human performance during the game, we decomposed each major step of the workflow and mapped it to instrumentation data and performance metrics that characterize players' behaviors. As noted in Figure 6, three major facets of performance emerged: client interaction, information triage, and discovery and decision. Additionally, the performance of this entire workflow is underpinned by a team's ability to effectively organize and collaborate through face-to-face communication. Our case study of four five-member teams illustrates how system-level and physiological instrumentation can be used to better characterize a team's performance during gameplay.

Game Client Interaction Performance

Software instrumentation built into Bluestreak recorded various user interactions both on demand and at specific intervals. The recorded data can be used to understand macro behaviors, such as the volume or rate of interactions with specific tools in the client. For example, by recording placemark creation and modification attributes, we can quantify team analytical behaviors in the workflow as a function of time. These data can also be used to analyze micro behaviors, such as a user's current look at geospatial data, known as the viewport [9]. Viewport data are recorded each second and include the current time of gameplay, the time in the scenario being displayed, and the geospatial bounding box of the video footprint in the map section of Bluestreak. An example of viewport instrumentation is shown in Figure 15.

Scenario Information Triage Performance

After the viewport data were logged, they were correlated with the scenario ground truth and processed using specialized information theoretic metrics [9, 21] to determine which relevant (scenario network) and irrelevant (background population) tracks or sites were being viewed at each scenario time step.

A graphical representation of the scenario and background track information, shown in Figure 16a, was used to assess a team's ability to effectively triage vehicle track data. If players were properly interpreting the information in the report messages, they should have focused only on the red scenario vehicle tracks and not the yellow background population tracks. As shown in Figure 16b, the performance of Team 3 and Team 4 plateaued as the scenario evolved, whereas Team 1 and Team 2 continued to find and analyze more relevant (red) scenario tracks throughout gameplay.

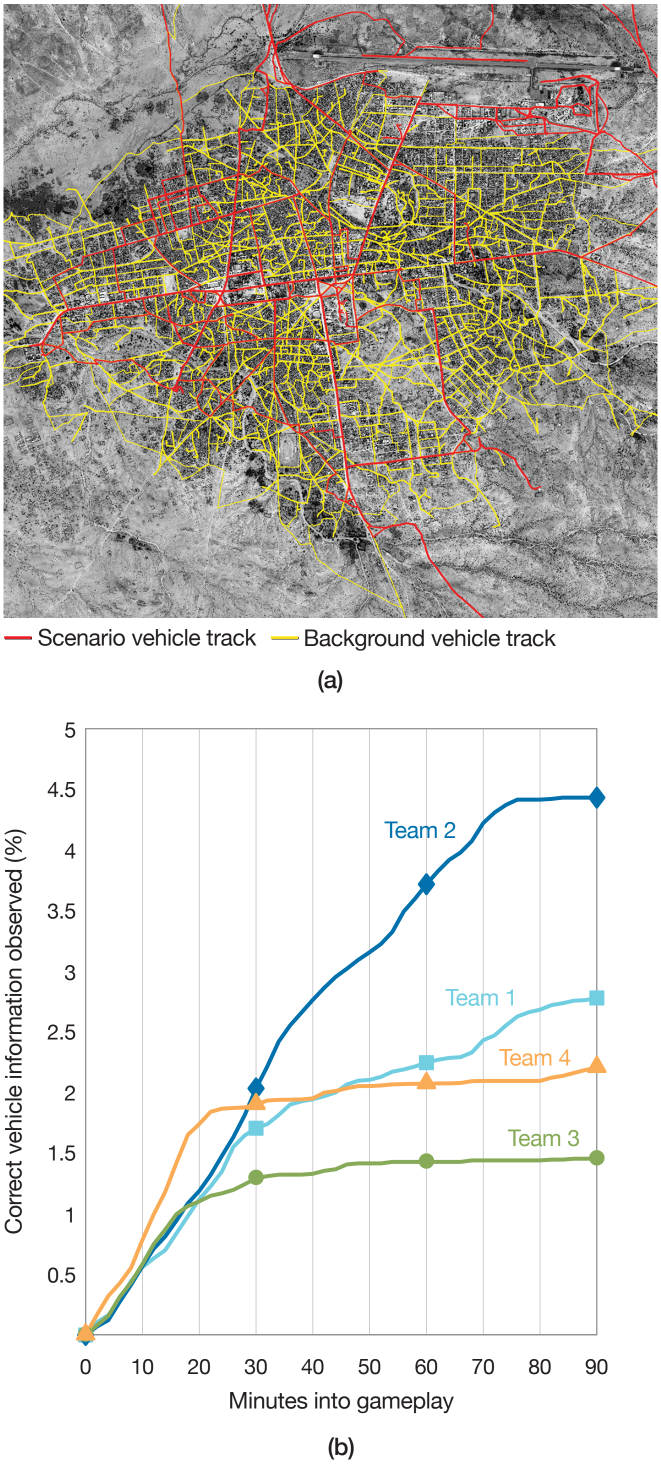
A graphical representation of the scenario red sites is shown in Figure 17a. If players are properly interpreting the information in the report messages, they should focus only on vehicle behaviors and activities around the sites with the red icons and not on the ones denoted with yellow dots that indicate locations of the background population not associated with the red network.

Similarly, Figure 17b illustrates teams' ability to effectively triage video of site-related activities. As the figure shows, Team 1 and Team 2 spent substantially more effort observing scenario site information compared to Team 3 and Team 4. In many cases, teams spent a lot of time analyzing sites but ultimately chose an incorrect action or took no action at all.

Team Discovery and Decision-Making Performance

Because the scenario was constructed to have the scenario activities completely separated from the background activity, the game can be analyzed from the perspective of signal detection theory. Essentially, teams can be considered detectors of scenario network activity in that they are attempting to extract these signals from the noise of the normal activities of the rest of the population [10]. The receiver operating characteristics (ROC) measurements of detection theory can be used to assess the teams' performance (Figure 18).

Results from two different tasks are plotted: the discovery of scenario sites, which is measured by team placemarks at those sites, and the declaration of scenario sites, which is the subset of the total placemarks that are assigned a course-of-action decision. Decision actions are



(a)

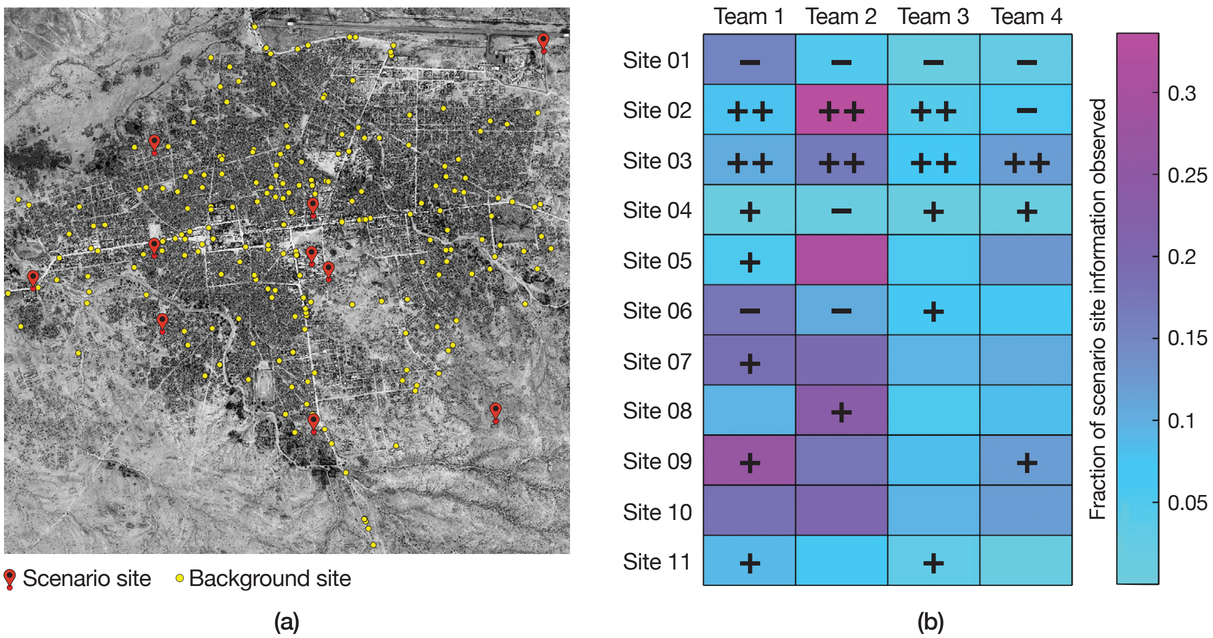
(b)

FIGURE 16. Team vehicle triage performance is depicted in the two plots. The plot in (a) shows the extents of all vehicle track data in the game, with the red lines denoting tracks associated with the scenario network vehicles and the yellow denoting tracks of background population tracks. The plot in (b) shows the team triage performance, with the *y*-axis representing the percentage of total red tracks observed in the video and the *x*-axis representing the number of minutes elapsed since the start of the game.

directly related to the teams' comprehension of the scenario and their confidence in that understanding. For example, it can be seen that Team 2 had placemarks on 100 percent of the scenario sites but only had the confidence to declare 30 percent of those sites. They also declared sites not part of the network, resulting in a 0.2 percent probability of false declaration. Team 4 had discovery performance similar to that of Team 1 and zero probability of false declaration. Team 1 had the highest detection probability but at the expense of more false declarations.

Team Verbal Communication Performance

Face-to-face communication is a key factor in overall team performance for highly cooperative tasks [22–25].



	Team 1	Team 2	Team 3	Team 4
Site 01	−	−	−	−
Site 02	++	++	++	−
Site 03	++	++	++	++
Site 04	+	−	+	+
Site 05	+			
Site 06	−	−	+	
Site 07	+			
Site 08		+		
Site 09	+			+
Site 10				
Site 11	+		+	

FIGURE 17. Team site triage performance is depicted. The plot in (a) shows the scenario sites to be discovered, annotated with red icons, and the background sites, denoted with yellow dots. The table in (b) shows teams' performance at accumulating information at each of the scenario sites, indicated by the fill color of each box and color bar scale. Decision outcomes for sites are also plotted in (b), with a + or – representing a correct or incorrect decision, respectively.

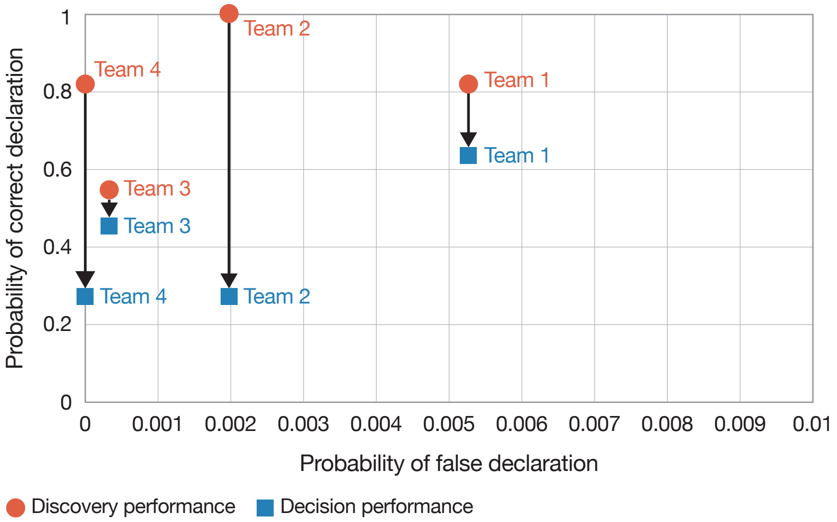


FIGURE 18. In this receiver operating characteristics plot, the *y*-axis represents the probability of correct declaration, or the fraction of correct sites found and acted upon by the teams, and the *x*-axis represents the probability of false declaration, or the ratio of incorrect sites declared divided by the total possible discoverable sites. The blue squares represent decision performance for sites that were declared to be associated with the network. The red circles show the fraction of all sites that were correctly discovered before the course-of-action selection process. The black arrows show the amount of performance lost moving from the information discovery process to the decision process, with the amount of performance loss a factor of each team's certainty about their understanding of the scenario, their risk tolerance, and their approach to making decisions.

Traditional methods to characterize these communications have largely focused on speech content; however, more recent methods center on the collection of nonlinguistic speech features that enable the characterization of team dynamics without having to analyze the linguistic content of a team's utterances [18, 23].

To collect speech metadata, we gave sociometric badges to each player during gameplay (Figure 19). The badges continuously recorded the time, duration, and identity of each player's speech, and post-processing software provided measurements of when a player spoke alone, when speech overlapped with another player, which players were listening, and when players were silent. These data naturally formed a directed graph of communication between players (Figure 20). For simplicity, graphs for only Team 1 and Team 2 are provided here.

Previous studies of face-to-face communication behaviors of small teams in a collaborative setting have found that balanced participation and speaking time along with increased turn-taking are associated with better team performance [26]. In Figure 20, Team 1 players A and C are dominating the conversation, as seen by their edge thicknesses, while the rest of the players are

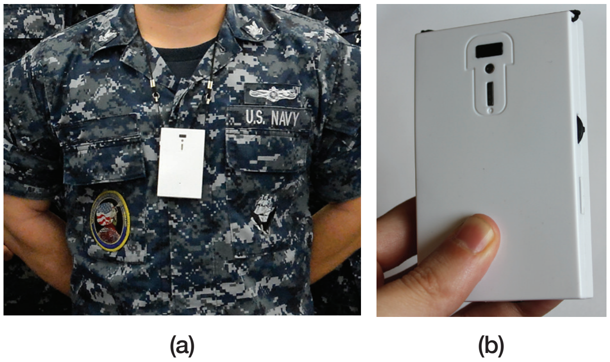


FIGURE 19. A U.S. Navy service member wears a sensor called a sociometric badge (a) that can record nonlinguistic metadata of speech behaviors, body movement, and other data. The battery-powered badge (b) incorporates a number of sensors, including a microphone, wireless and infrared transceivers, and a three-axis accelerometer. Microphones combined with specialized filters and signal processing characterize when the wearer is speaking. Wireless and infrared transceivers allow the badges to identify other badges proximal to them, and when data from the nearby badges are combined with the speech data, the communication patterns of who speaks to whom within a team can be determined. This directed speaking data can be used to measure team-based speech behaviors, such as turn-taking and interruptions. The accelerometer data can determine features associated with excitement and engagement.

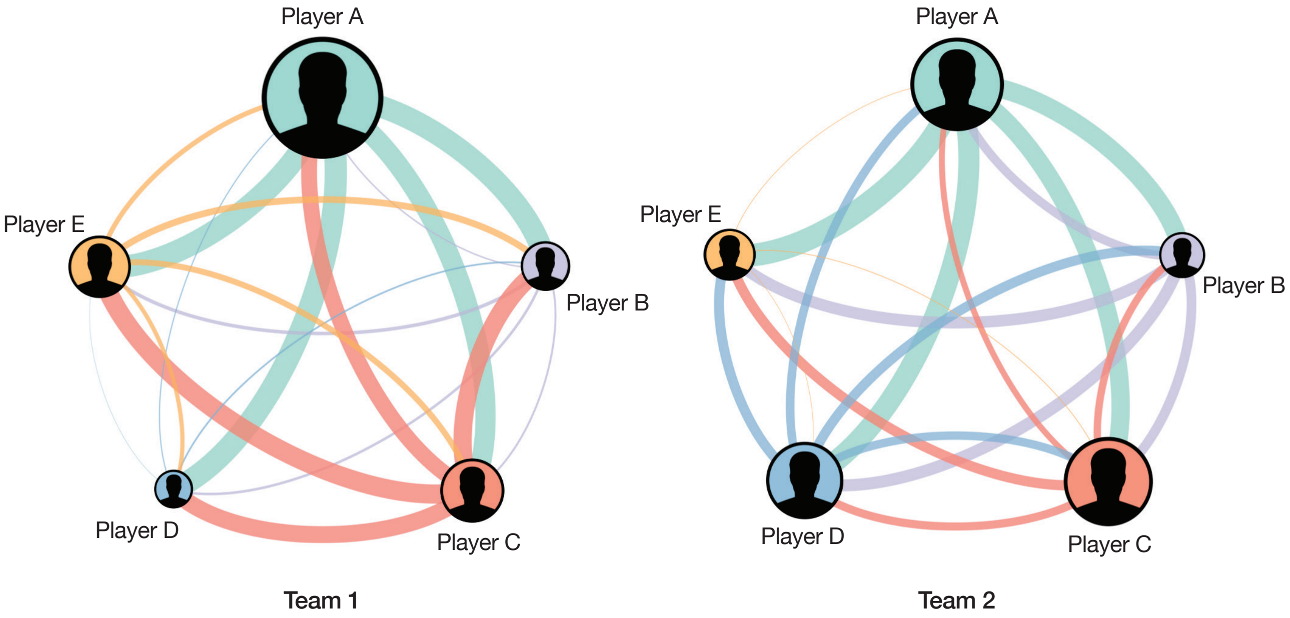


FIGURE 20. The graphs illustrate face-to-face communication networks. Vertices (circles) represent players and edges (lines) represent directed communication from one player to another. Vertex size is proportional to total participation for a player, edge thickness is proportional to directed speech time to each teammate, and edge color indicates directionality by matching the source vertex color.

less engaged with lower participation (smaller vertices) and less speaking time (thinner edges). Conversely, Team 2 has a much more balanced distribution of both speaking time and participation than Team 1, with player A acting in the role of team leader. Analysis that uses such graph data is a promising area of active research into group influencers and team role estimation [27].

For deeper insight into the communication network, we explored a social network analysis approach to characterizing player interaction. By computing the directed, normalized closeness centrality of each player [28], we can derive an estimate of the connectedness of players. Larger centrality magnitudes indicate a player's graph closeness to all other players. One useful application of this measure is to inspect the time-varying behavior of player centrality [29] during gameplay (Figure 21). In the figure, the visual representation of Team 1's and Team 2's closeness centrality can be useful for identifying team dynamics, such as the emergence of a leader. In Team 1, we see the same communication dominance exhibited by players A and C as seen in Figure 20. In Team 2, player A clearly emerges as the leader during the discovery phase of the game, with A's centrality decreasing toward the end when the team moved into the collective decision-making phase of the game.

In addition to performing a social network analysis, we did a recurrent pattern analysis that used the data collected by the sociometric badges. First, speech patterns were coded into symbols according to various speech behaviors and then analyzed as a time series [30]. The strength of the recurrent structure within these code sequences is called determinism (DET). In a strict turn-taking situation, DET will be high (near 100 percent) as the conversation is highly structured. In a situation with random speech intervals, DET will be low (close to 0 percent), indicating that the conversation is highly unstructured. DET scores were comparable for the four teams, with local maxima near 60 percent and local minima near 30 percent. Fluctuations in the values occurred over time, indicating that the structure of the communication ebbed and flowed throughout gameplay. Further analysis showed a high correlation between DET magnitude and the percentage of time an individual spoke while all others listened, suggesting that structure occurs, even in a complex team setting with five participants, when individuals speak and others listen.

Total Team Performance

We quantitatively measured team performance at several points in the overall game workflow. However, combining these metrics into a single total performance measure

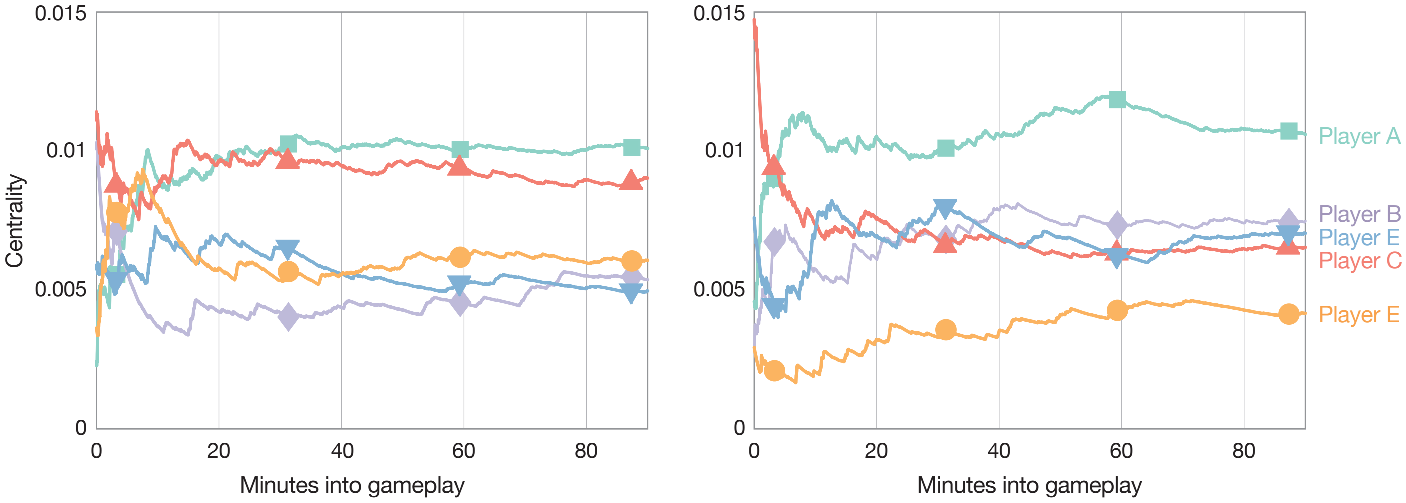


FIGURE 21. The plots depict the time-varying player communication centrality. Centrality is a social network analysis measure that can be used to identify the most important vertices in a network. The directed normalized closeness centrality of each player is an estimate of the connectedness of players in a network. The *x*-axis represents elapsed time during gameplay, and the *y*-axis represents the centrality of a player. Larger centrality magnitudes indicate a player's graph closeness to all other players. In both teams, player A is considered the leader and transitions to gain the highest centrality midway through the game. Qualitative observations during gameplay supported these findings.

warrants careful consideration. Qualitatively, Team 1 and Team 2 excelled at communication, triage, and site discovery, but they had more false declarations than Team 3 or Team 4. Conversely, although Team 3 and Team 4 did not observe as much information or discover as many sites as did Team 1 and Team 2, they were very accurate in adjudicating what they found. Team 2 ultimately won the four-team competition with the best overall performance and game scores.

Predicting Team Performance

When we assessed teams' analytical and decision-making performance, common questions arose regarding how performance in one facet of a decision process affects the performance of either subsequent processes or the aggregate overall process. The previous sections illustrate that the collected measurements enabled detailed insight about individual facets of performance; however, we wanted to take this a step further to determine whether behaviors in specific facets of the intra-game workflow were predictive of analytical performance of players or the outcomes of games. To approach this investigation, we processed data collected over several years of gameplay, encompassing 71 different teams and more than 350 unique players. For all 71 teams, system instrumentation data were recorded. For a subset of 15 teams, face-to-face communication data were also collected.

Robust linear regression analyses were used to statistically estimate how predictive were the various facets of intra-game performance with respect to workflow processes. For each model, residual analysis, significance testing, and other regression diagnostics were performed, and were evaluated for each prediction finding. In undertaking this analysis, we wanted to address three overarching research propositions:

- Client interaction effectiveness. The first proposition asked whether more effective interaction with the game software client led to better game performance. From our analysis, we found that teams who had higher usage across all analytic functions of the game client discovered more total sites and had a higher probability of correct site discovery. The effect was even more pronounced for the functions of the game client associated with the frequency with which players submitted Nomination space-time queries for track data and its correlation with increased site discovery. Higher total game client interaction was also associated with more effective observations of scenario site information and track information. Essentially, teams who were more effective at interacting with the functions of the game client observed more relevant scenario information and found more correct sites.
- Information triage effectiveness. The second proposition asked whether discovery of more scenario information led to better game outcomes. From our analysis, we found that teams who observed more relevant scenario site information and track information also scored higher in game outcome. The overall game score takes into account several aspects of how well the players perform, but it also encapsulates the confidence of players' decisions (course–of-action strength) and reflects their overall strategy for the game (aggressive to risk averse).
- Team communication effectiveness. The third proposition asked whether teams who communicate more effectively have higher game performance. Our analysis found that teams who communicated more (total time) throughout the exercise also observed more relevant scenario site information and track information. Additionally, teams who had higher participation (frequency of communication) from all members throughout the game also observed more relevant scenario site information and track information. Lastly, teams who communicated more (total time) throughout the exercise also made better decisions on the most challenging sites to adjudicate. These findings about total team engagement and participation agree with our qualitative observations of teams during the decision-making process. Team centrality metrics did not have a significant association with other aspects of team performance and warrant further investigation.

Follow-on Work

The concepts explored during this work and the lessons learned yielded two major accomplishments. The first included the expansion of the Humatics instrumentation framework to take in additional sources and types of data, the development of new methods for real-time and post-exercises metrics and assessment visualizations, and a series of research efforts focused on a better understanding of analytical performance.

The second major consequence was the production of two serious games that followed a set of development and employment mechanisms that were similar to those we used in our work. One game focused on an airport security scenario in which teams who had access to actual closed-circuit video from a major U.S. airport monitored the video and other data feeds to discover suspicious activities being performed by scripted actors. The second game involved all-source information analysis during which participants analyzed documents, answered questions, and made recommendations regarding a complex geopolitical event while intricate human-system interaction data were collected with a high-frame-rate, near-infrared, eye-tracking system and a custom instrumented instance of the Palantir Technologies data analysis platform. This latter game focused on a detailed user-workflow decomposition and metrics development to characterize individuals' reading behaviors, estimate their cognitive load, and objectively assess their performance at information discovery, factual recall, inference development, and decision making.

Acknowledgments

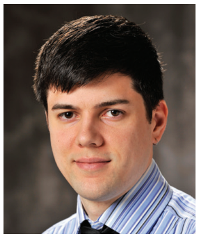
The authors would like to thank Christian Anderson, Joaquin Avellan, Peggy Chernoff, John Collins, Gary Condon, Jason Hepp, Benjamin Landon, Rebecca Madsen, Joseph Marini, Kyle O'Brien, Travis Riley, Timothy Schreiner, Kenneth Senne, Bryan Tipton, Andrew Wang, and James Won for their contributions to this research. ■

References

1. D.A. Bright, C.E. Hughes, and J. Chalmers, "Illuminating Dark Networks: A Social Network Analysis of an Australian Drug Trafficking Syndicate," *Crime, Law and Social Change*, vol. 57, no. 2, 2012, pp. 151–176.
2. F. Calderoni, "The Structure of Drug Trafficking Mafias: The 'Ndrangheta and Cocaine," *Crime, Law and Social Change*, vol. 58, no. 3, 2012, pp. 321–349.
3. R.M. Bakker, J. Raab, and H.B. Milward, "A Preliminary Theory of Dark Network Resilience," *Journal of Policy Analysis and Management*, vol. 31, no. 1, 2012, pp. 33–62.
4. P.K. Davis and K. Cragin, eds., *Social Science for Counterterrorism: Putting the Pieces Together*. Santa Monica, Calif.: Rand Corporation, 2009.
5. J.N. Shapiro, *The Terrorist's Dilemma: Managing Violent Covert Organizations*. Princeton, N.J.: Princeton University Press, 2013.
6. S. Helfstein and D. Wright, "Covert or Convenient? Evolution of Terror Attack Networks," *Journal of Conflict Resolution*, vol. 55, no. 5, 2011, pp. 785– 813.
7. "Car Bomb," Wikipedia, accessed online 17 July 2018, https://en.wikipedia.org/wiki/Car bomb.
8. A.H. Tapia, N.J. LaLone, and H.-W. Kim, "Run Amok: Group Crowd Participation in Identifying the Bomb and Bomber from the Boston Marathon Bombing," in *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management*, 2014, pp. 265–274.
9. M. Daggett, K. O'Brien, and M. Hurley, "An Information Theoretic Approach for Measuring Data Discovery and Utilization during Analytical and Decision-Making Processes," in A. De Gloria and R. Veltkamp, eds. *Games and Learning Alliance, Lecture Notes in Computer Science*, vol. 9599. Basel, Switzerland: Springer, 2015.
10. J.C. Won, "Influence of Resource Allocation on Teamwork and Performance in an Intelligence, Surveillance, and Reconnaissance (ISR) Red/Blue Exercise within Self-Organizing Teams," PhD dissertation, Tufts University, 2012.
11. J.C. Won, G.R. Condon, B.R. Landon, A.R. Wang, and D.J. Hannon, "Assessing Team Workload and Situational Awareness in an Intelligence, Surveillance, and Reconnaissance (ISR) Simulation Exercise," in *Proceedings of the IEEE First International Multi-disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, 2011, pp. 163–167.
12. R. Porter, A.M. Fraser, and D. Hush, "Wide-Area Motion Imagery," *IEEE Signal Processing Magazine*, vol. 27, no. 5, 2010, pp. 56–65.
13. D.G. Bell, F. Kuehnel, C. Maxwell, R. Kim, K. Kasraie, T. Gaskins, P. Hogan, and J. Coughlan, "NASA World Wind: Opensource GIS for Mission Operations," in *Proceedings of the 2007 IEEE Aerospace Conference*, 2007, pp. 1–9.
14. R.R. Vatsavai, S. Shekhar, T.E. Burk, and S. Lime, "UMN-Mapserver: A High-Performance, Interoperable, and Open Source Web Mapping and Geo-spatial Analysis System," in M. Raubal, H.J Miller, A.U. Frank, and M.F. Goodchild, eds. *Geographic Information Science. Lecture Notes in Computer Science*, vol. 4197. Berlin & Heidelberg, Germany: Springer, 2006, pp. 400–417.
15. M.R. Endsley, "Situation Awareness Misconceptions and Misunderstandings," *Journal of Cognitive Engineering and Decision Making*, vol. 9, no. 1, 2015, pp. 4–32.
16. P. Salmon, N. Stanton, G. Walker, and D. Green, "Situation Awareness Measurement: A Review of Applicability for C4i Environments," *Applied Ergonomics*, vol. 37, no. 2, 2006, pp. 225–238.
17. J. Klingner, R. Kumar, and P. Hanrahan, "Measuring the Task-Evoked Pupillary Response with a Remote Eye Tracker," in *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, 2008, pp. 69–72.
18. D. Olguín-Olguín and A. Pentland, "Sensor-Based Organizational Design and Engineering," *International Journal of Organisational Design and Engineering*, vol. 1, no. 1–2, 2010, pp. 69–97.

19. J.J. Garrett, *The Elements of User Experience: User-Centered Design for the Web and Beyond*. Berkeley, Calif.: Pearson Education, 2010.
20. M. Daggett, K. O'Brien, M. Hurley, and D. Hannon, "Predicting Team Performance Through Human Behavioral Sensing and Quantitative Workflow Instrumentation," in I. Nunes, ed., *Advances in Human Factors and System Interactions. Advances in Intelligent Systems and Computing*, vol. 497. Basel, Switzerland: Springer, 2017, pp. 245–258.
21. E.K. Kao, M.P. Daggett, and M.B. Hurley, "An Information Theoretic Approach for Tracker Performance Evaluation," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 1523–1529.
22. X.S. Apedoe, K.V. Mattis, B. Rowden-Quince, and C.D. Schunn, "Examining the Role of Verbal Interaction in Team Success on a Design Challenge," in *Proceedings of the 9th International Conference of the Learning Sciences*, vol. 1, 2010, pp. 596–603.
23. A.J. Strang, S. Horwood, C. Best, G.J. Funke, B.A. Knott, and S.M. Russell, "Examining Temporal Regularity in Categorical Team Communication Using Sample Entropy," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, no. 1, 2012, pp. 473–477.
24. L.A. Whitaker, S.L. Fox, and L.J. Peters, "Communication Between Crews: The Effects of Speech Intelligibility on Team Performance," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 37, no. 9, 1993, pp. 630–634.
25. H.P. Andres, "The Impact of Communication Medium on Virtual Team Group Process," *Information Resources Management Journal*, vol. 19, no. 2, 2006, pp. 1–17.
26. W. Dong, B. Lepri, T. Kim, F. Pianesi, and A.S. Pentland, "Modeling Conversational Dynamics and Performance in a Social Dilemma Task," in *Proceedings of the 5th International Symposium on Communications, Control and Signal Processing*, 2012, pp. 1–4.
27. W. Dong, B. Lepri, A. Cappelletti, A.S. Pentland, F. Pianesi, and M. Zancanaro, "Using the Influence Model to Recognize Functional Roles in Meetings," in *Proceedings of the 9th International Conference on Multimodal Interfaces*, 2007, pp. 271–278.
28. L.C. Freeman, D. Roeder, and R.R. Mulholland, "Centrality in Social Networks: II. Experimental Results," *Social Networks*, vol. 2, no. 2, 1979–1980, pp. 119–141.
29. K. Ara, N. Kanehira, D. Olguín-Olguín, B.N. Waber, T. Kim, A. Mohan, et al., "Sensible Organizations: Changing Our Businesses and Work Styles Through Sensor Data," *Journal of Information Processing*, vol. 16, 2008, pp. 1–12.
30. J.C. Gorman, N.J. Cooke, P.G. Amazeen, and S. Fouse, "Measuring Patterns in Team Interaction Sequences Using a Discrete Recurrence Approach," *Human Factors*, vol. 54, no. 4, 2012, pp. 503–517.

About the Authors



Matthew P. Daggett is a member of the technical staff in the Humanitarian Assistance and Disaster Relief Systems Group. He joined Lincoln Laboratory in 2005, and his research focuses on using operations research methodologies and quantitative human-system instrumentation to design and measure the effectiveness of analytic technologies and processes for complex sociotechnical systems. He has expertise in remote sensing optimization, social network analysis, natural-language processing, data visualization, and the study of team dynamics and decision making. He holds a bachelor's degree in electrical engineering from Virginia Polytechnic Institute and State University.

Daniel J. Hannon is a research psychologist and a member of the technical staff in Lincoln Laboratory's Bioengineering Systems and Technology Group. His work spans interests in psychological health, cognitive science, teamwork, and human factors engineering. In addition to his research career, he is an active clinician, working in emergency psychological care and also teaches in the Mechanical Engineering Department at Tufts University. Prior to joining the Laboratory, he was the director of the Human Factors Engineering Program at Tufts University and a program manager in Aviation Human Factors at the U.S. Department of Transportation, Volpe Center. He holds a bachelor's degree in psychology from Nazareth College and master's and doctoral degrees in experimental psychology from Brown University.

Michael B. Hurley is a member of the technical staff in the Intelligence and Decision Technologies Group at Lincoln Laboratory. Over his career at the Laboratory, he has worked on projects involving real-time servo control systems, multitarget tracking, multisensor data fusion, and probabilistic and information theoretic methods for assessing performance. His current research interest is the use of Bayesian probability theory and information theory to design decision support algorithms. He holds a bachelor's degree in physics from Carnegie-Mellon University and a doctorate in physics from the University of Pennsylvania, where his graduate work was in neutrino physics.

John O. Nwagbaraocha is an assistant group leader of the Embedded and Open Systems Group, where he focuses on service-oriented architectures to address challenging national security problems. In his previous role as a technical staff member, he led teams in the development of reference architectures and prototypes for the Air Force and the Intelligence Community. He joined the Laboratory in 2007 and worked on synthetic aperture radar processing, activity-based analytics for moving target indicator data, and user interface design for serious games. He holds a bachelor's degree in computer engineering from the Rochester Institute of Technology and a master's degree in electrical engineering from Northeastern University.

Early Gaming at Lincoln Laboratory: The Missile Defense Engagement Exercises of 1966 to 1968

Researchers worked through the operational logic of a complex defense system in the early years of U.S. missile defense research.

What brought Lincoln Laboratory into missile defense research? The Laboratory was established in the early 1950s to develop a continental air defense against Soviet bombers carrying nuclear weapons. The architecture of this air defense system, developed under U.S. Air Force leadership, featured a wide deployment of radars to detect and track attacking bombers, and fighter interceptors to engage and destroy the enemy aircraft. This architecture was essentially a defense of the full area of the United States and Canada.

A different architecture was favored by the U.S. Army and its major development arm, the prestigious Bell Telephone Laboratories. Their architecture, referred to as the Nike Ajax System, featured a localized defense around major cities with radar sensors and guided-missile interceptors. The extreme concern in the United States concerning nuclear attacks led to both architectures being deployed, and by the 1960s the air defense of the United States and Canada comprised a truly massive system.

The late 1950s development of long-range ballistic missiles capable of delivering nuclear warheads to intercontinental distances began to shift the nation's concern away from air defense toward missile defense. The U.S. Army had the lead role in missile defense and, together with Bell Laboratories, conducted a successful intercept of an intercontinental ballistic missile (ICBM) target at Kwajalein Atoll of the Marshall Island in 1962.

In this same era, Lincoln Laboratory became involved in systems to warn of ballistic missile attack, performing architecture work on the Ballistic Missile Early Warning System (BMEWS) that became operational in early 1964. This work naturally led the Laboratory to consider the technological challenges of ballistic missile defense.

Some of the leadership in the Department of Defense thought the Army–Bell Laboratories approach to ballistic missile defense embodied in the Nike-X system was unduly conservative. The technology of ballistic missiles was improving rapidly and the department encouraged projects that were technologically more advanced than the Army's Nike-X program.

The Laboratory entered the missile defense domain in the early 1960s with experiments designed to capture the physics of a missile warhead reentering Earth's atmosphere at hypersonic speeds. This "reentry physics" effort focused on how to distinguish a real warhead from a wide variety of debris from the parent rocket and possibly countermeasure devices such as decoys. Experiments began at Wallops Island, Virginia, then migrated to the White Sands Missile Range, New Mexico, and finally to the Kwajalein Atoll in the Pacific in 1962. This reentry physics challenge was daunting. All we needed to do was weigh objects at a substantial distance (100 km) by "tickling" them with a radar beam! The objects are moving at greater than 20,000 feet per second. They are decelerating at a peak of 60 gs, and they may have an ionized trail attached. We need to do this weighing process in a few seconds, possibly on a number of objects—a heroic challenge, but an intriguing one!

The Lincoln Laboratory Effort

Considerable controversy has surrounded missile defense since its inception: "hitting a bullet with a bullet" was judged too difficult in those early days. A

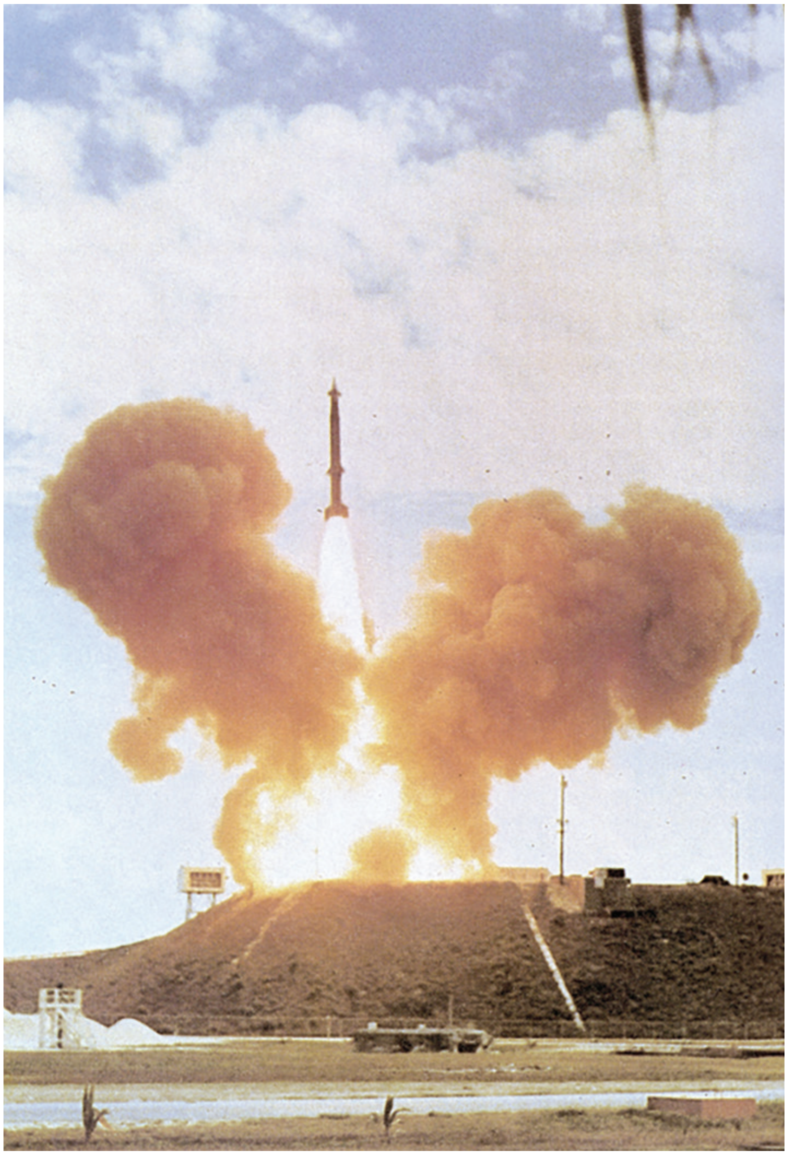


Photo: William Delaney

(a)

Photo: U.S. Army

(b)

The Nike-X interceptors: These two high-performance interceptor missiles (produced by McDonnell Douglas and the Martin Marietta Company) were the backbone of U.S. missile defense research and development in the 1960s and 1970s. They featured high speed and high acceleration, and their launches were spectacular. The author had a box-seat for the first Spartan launch shown in photo (a) at Kwajalein in 1968. The Sprint missile in photo (b) was launched from White Sands Missile Range, New Mexico. These interceptors became major components in the Safeguard missile defense of 1975.

missile defense system must function almost completely automatically; there is not enough time in the engagement of a ballistic missile for a lot of human control and decision making. Skeptics in that era, and even today, believe the necessity for a rapid and flawless execution of an engagement logic is one of the big impossibilities in missile defense.

We researchers at Lincoln Laboratory were intensely curious as to how much of this automated engagement logic had been worked out by Bell Labs for their Nike-X urban defense architecture. The Bell Labs scientists alluded to work on the topic but never presented any results. We suspected that they had not gotten very far on that problem. So, we began to look at the rough elements of a logic that would be needed to launch an interceptor at some incoming ICBM warhead and not launch interceptors at the various pieces of missile hardware junk or countermeasures that could accompany the warheads. These early missile defense systems did their most confident defense in the atmosphere, so there were many reentry physics logic questions to answer. We used models of the Nike-X interceptors in our work but posed new radar models with more advanced capabilities than the Nike radars.

This engagement logic work was done in the Radar Division under the calm leadership of Donald Clark, who was convinced that the engagement logic question was critical to missile defense. The less calm intellectual lead on missile defense systems was Joel Resnick, and the principal system-oriented staff members were John Fielding, Stephen Weiner, and this author.

The Engagement Logic Development and Gaming

Putting together the computer-logic flow for a missile defense system was a challenging task. No one had done it before, but we bravely marched in. How to test one's logic became a prominent question, and we evolved the "Engagement Exercises" as a gaming process to test our logic. An exercise was a bit like our current "red-blue" discrimination games that challenge one set of participants (red team) to devise methods to prevent a defender from knowing which of many objects around an attacking ICBM complex is a real warhead and another set (blue team) to determine strategies to discriminate the missile from decoys or debris. However, our scope was much broader than just discrimination. We featured the whole set of surveillance, detection, verification, tracking, discrimination, interceptor commitment, and guidance processes. We started simple, with simple offense-defense scenarios and built up to more complex games over the course of three years.

The archives show that our first exercise was in May 1966, and exercises followed at roughly six-month intervals for a total of six exercises until the last one in early 1968. We would work for six months preparing the defense logic, which was quite detailed with numerical thresholds for the initiation of some defense process or some defense identification of an object. The setting of numerical thresholds on all processes was a challenge. This "defense team" was opposed by an "offense team" that conjured up a missile threat in gory detail. We were isolated from each other, and the secrecy was tight.

Overseeing both defense and offense teams was the "umpire team" that set ground rules on how much knowledge the opposing teams had of each other (mimicking the information gathered by intelligence communities) and generally inspecting both teams' work for completeness and fairness. The umpire team was a major force in making things proceed in a logical and productive manner, and when we met for the engagement exercise, the umpires were very much in charge. I recall that John Fielding often chaired the umpire team, and that role suited him very well. He assumed a somewhat imperious style, a bit like that of a judge. He coined the phrase "social stigma" as the presumed penalty for overstatements of capability by the defense or offense as we engaged each other. Our group leader Don Clark was often an umpire and his aura of total fairness helped keep things calm.

> We were informing ourselves and our sponsor on the complexities of missile defense warfare—and that was good training for the Laboratory's ensuing 50+ years in missile defense.

I was always a defender as was Joel Resnick. The very creative Bob Bergemann of the Data Systems Division and Dave Towle of the Radar Division were professional offense team leaders. A few supporting organization were involved with us. The Cornell Aeronautical Laboratory in Ithaca, New York, and the Kaman Nuclear Corporation of Albuquerque, New Mexico, provided support in threat modeling. A dominant contribution came from the Defense Research Corporation (DRC), later named the General Research Corporation (GRC), of Santa Barbara, California; they were building a huge computer representation of a ballistic missile engagement and had many useful tools, such as trajectory generators and interceptor missile fly-out trajectories.

The DRC team was a collection of very smart "West Coasters" who were intensely interested in our work because Lincoln Laboratory had lots of real-world depth in discrimination, tracking, detection, and false-alarm mitigation. The Laboratory's experience from Kwajalein and many air defense hardware efforts was a great complement to their predominant computer simulation expertise. They became a major player in this work, and we felt a great deal of satisfaction in having two great teams with a shared vision. Jack Ballantine led the DRC team; he was a lot like the Laboratory's Don Clark in calm demeanor. The DRC team's "California cool" was a good offset to the East Coast aggressive styles of Resnick, Fielding, and Delaney.

The Engagement Exercises (Game)

The engagement exercises each took a full three days. They were conducted in a somewhat formal manner, much like a courtroom. The defense team, accompanied by a pile of large paper drawings of the "defense logic," sat in their designated area in a big room. The offense team, armed with their technical documentation of their "threat," did the same.

The umpires sat in a central position. The urban defense system for the United States had been specified well in advance by the defense. The umpires would start with a statement on the world situation, an input on the state of the Soviet Union, and any warning indicators. Then, for example, they might tell us that an Alaskan BMEWS radar was down for repair.

The action would begin when the umpires announced that the BMEWS radar at Thule, Greenland, had received signal return from some object at such-and-such a range and angle and asked the defense, "What do you do next?" Our logic would call for a verify transmission and then a velocity estimate to see if the detection was caused by a satellite or a missile. If the target report passed our missile thresholds, we would send out additional pulses and then follow our logic train of crude impact-point determination, handover to a tracker, track to refine an intercept point, followed by an intercept process. But, things never went that smoothly. At the first engagement exercise, we could not get anything logical to happen in response to our repeated attempts to start a target track or predict an impact point. Eventually, after several hours of tortuous debate and argument with the umpires, they confessed to giving us highly range-ambiguous returns from the moon as our first target (mirroring a real-world event with BMEWS).

Developing a defense logic was a complicated process, even for simple threats, and along the way we noted many shortcomings in our logic. Our leader, Don Clark, would remind us that our goal was to find those shortfalls, and while we intellectually agreed, we defenders wanted to win!

Eventually, our exercises attracted an audience beyond the participants. I recall sometimes acting more like a defense lawyer and doing a bit of showboating along the way. On one such exercise, I had Lincoln Laboratory's Kent Kresa as my cochair on the defense team (Kent went on to a most impressive career, culminating in a position as CEO of Northrop Grumman). On the third day of the exercise, the defense logic was beginning to ferret out the real warheads to be intercepted in a background of countermeasures and interference, and we were launching our Sprint interceptors left and right per our logic. Kent came up and put a Red Auerbach cigar in my mouth, lit it for me, and said, "We beat these guys!" So there was a spirit of winning that kept us on our toes throughout this six-month process we called an exercise.

We continued to conduct these exercises, each with a six-month preparation, over three years, and the game became increasingly complex as we dealt with countermeasures, such as chaff, decoys, jammers, and nuclear blackout generated by the offense or by our own defense interceptor bursts. As defenders, we were learning some tricks of our own, like precommitment of interceptors to provide early intercept options and shoot-look-shoot opportunities. We were finding out which radar capabilities made a big difference. We were also dealing with some nightmare scenarios involving huge enemy warheads that could destroy a city by bursting at very high altitudes, and we invented "the big bomb alarm" and defense logic to thwart that attack.

Overall, we were informing ourselves and our sponsor on the complexities of missile defense warfare. While our work did not appear directly in a system, we were teaching ourselves just how difficult the missile defense job might be, and that was good training for the Laboratory's ensuing 50+ years of work in missile defense. I claim we were the first in the nation to take a hard look at this daunting missile defense engagement logic problem and test ourselves with a gaming process. I am proud to have been part of that fine team of talent.

— WILLIAM DELANEY

Bill is a veteran of 61 years at the Laboratory. He is currently the Director's Office Fellow and is a former Assistant Director. He spent many years in missile defense activities with a tour at the Kwajalein test site and a tour in the Office of the Secretary of Defense with responsibilities for missile defense research and development.

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
244 Wood Street ▪ Lexington, Massachusetts 02421-6426

www.ll.mit.edu

LINCOLN LABORATORY *JOURNAL*

Volume 23 ▪ Number 1 ▪ 2019