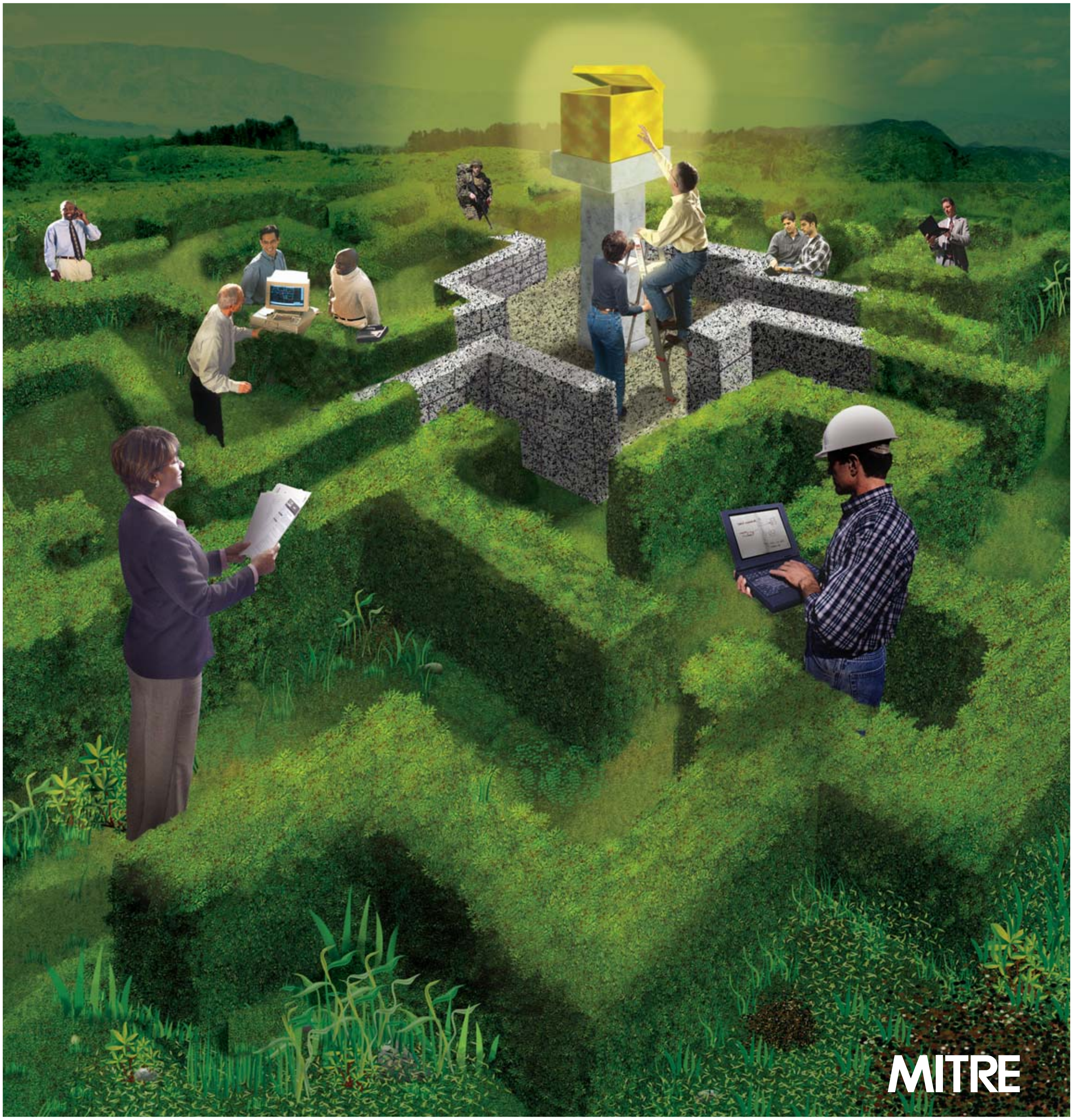
Experimentation and Prototyping

THE EDGE

MITRE's Advanced Technology Newsletter

Fall 2004 | Volume 8 | Number 2

www.mitre.org/edge



MITRE

INTRODUCTION by *Glenn Roberts & Spurge Norman*

Guest Editors

The systems engineering problems our government sponsors face today are more challenging than ever. Understanding how to build and manage these complex systems often involves experimentation and prototyping to find not just a solution to the problem, but the most efficient and effective solution—one that all stakeholders can support.

MITRE's involvement in experimentation covers a wide range of activities, from small experiments in our own R&D facilities to participation in global military exercises involving multiple organizations. Advancements in computer technology, prototyping tools, and simulation and visualization capabilities enable diverse stakeholders to share a "real world" experience in a laboratory or in the field. Stakeholders can then discuss their hands-on perspective of a proposed capability or concept and provide feedback about its pros and cons. When it comes to understanding and agreeing on change, a picture may be worth a thousand words, but a prototype can be priceless!

Our prototyping efforts also cover a wide range, from creating a simple virtual prototype that helps people understand an approach to building a complex prototype that we will turn over to industry in a technology transfer. Prototyping provides invaluable experience to the developers. Only through building something do you truly understand the design tradeoffs and possibilities. Mistakes are a critical part of this learning process. (As Dr. Craig Lawrence of IDEO, a firm that helps companies innovate through design, says, "Fail early in order to succeed sooner.") It is through prototyping and experimentation that we are able to fully explore a range of possible solutions to a problem and move efficiently toward consensus-based agreement.

MITRE conducts and participates in experiments and creates prototypes for many reasons, including:

- Understanding user requirements
- Receiving operator feedback
- Advancing technology
- Determining performance feasibility
- Assessing alternatives
- Reducing risks
- Training MITRE staff in new capabilities

The articles in this issue highlight some of our work in experimentation and prototyping, our methods (e.g., the use of simulations and rapid prototyping), issues we encountered (technical and organizational), and solutions we have achieved for our various sponsors.

The *Edge* Online

This issue of the *Edge* is also available online at *www.mitre.org/edge*. Don't miss the article, "Prototyping Brings New Joint Forces Capabilities to the Field," found only in the online edition. In it we discuss the U.S. Joint Forces' focus on using prototypes to sort out the best ideas to take to the field.

For more information, contact Glenn Roberts, *groberts@mitre.org* or 703-883-6820, or Spurge Norman, *stn@mitre.org* or 703-883-5398.

In This Issue

Advancing Aviation Systems Through Prototyping and Simulation

By Urmila Hiremath and Glenn Roberts

They're back. This past summer airports in Atlanta, Chicago, New York, Philadelphia, and elsewhere began seeing some of the same high levels of flight delays that frustrated travelers and airlines prior to the 2001 industry downturn. Demand for air travel is up due to a resurging economy and the growth f a new breed of low-cost air carriers, and the long-term prognosis is for continued growth, with traffic levels expected to increase by 50 percent or more by the year 2020.

MITRE's Center for Advanced Aviation System Development (CAASD), the Federal Aviation Administration's (FAA's) federally funded research and development center, supports the FAA in its efforts to address these delay and capacity issues while at the same time ensuring safety and security. To do this we've built advanced analytical, simulation, visualization, and prototyping capabilities—designed specifically to address complex problems such as redesigning the National Airspace System (NAS) to reduce flight delays.

The advancement of NAS capabilities involves a coordinated set of investments and decisions among a diverse and distributed set of stakeholders. For example, to introduce new technology:

- The FAA may have to invest in capital equipment, procedures development, and workforce training.
- Aircraft and avionics manufacturers may have to develop and certify new equipment.
- Airlines may need to invest in the new equipment and in training for pilots and maintenance crews.
- Military services may also need to purchase equipment and invest in procedural changes and training.
- Municipal airport authorities may need to make capital investments in airport facilities and infrastructure.

Thus, to be successful, all new aviation tools and procedures require mutual acceptance by a wide range of players—short of situations requiring administrative or legislative mandate.

Before such a set of coordinated investment decisions can be made, the various stakeholders must share a common understanding of the improvement, the changes that it will entail, and the benefits that it will provide to each of them and the "system" as a whole. We have found that CAASD's labs are well positioned to bring together stakeholders and let them experience, in a realistic setting, how changes will play out. Participants can operate prototypes and discuss in real time what works and what doesn't, suggesting and agreeing on changes that move the development along.

In the world of aviation—where safety is a primary concern—new concepts, procedures, and systems cannot be introduced hastily. An evolutionary process with direct stakeholder involvement is required.

CAASD's integrated Air Traffic Management (ATM) Laboratory operates at a critical point on this path. The lab represents a realistic, mid-fidelity user experience—its simulations look and feel close to the real thing while also having a high degree of flexibility, enabling us to quickly change the underlying concepts and displays based on user feedback. This provides an ideal testbed for trying out revolutionary new concepts and ideas to see what is possible. With these investments we have established a rapid prototyping capability facilitated by a flexible real-time simulation architecture and a data-rich supporting infrastructure.

The Progression of an Idea

The core objective of the lab is to support concept exploration and development—enabling the

Continues on page 4

Continued from page 3

Advancing Aviation Through Prototyping and Simulation

spiral development of new concepts by modifying, enhancing, or integrating new capabilities into the lab, developing scenarios in which to examine these concepts, and developing procedures to support them. The combined expertise of CAASD staff members (including former pilots and controllers, human factors engineers, and software and simulation developers) keeps simulations realistic and up to date.

"Storyboarding" is often used as a first step. Tools such as PowerPoint and Macromedia Director allow engineers to sketch out the concept and understand basic requirements. Concept prototypes are then brought to life to allow hands-on demonstrations and evaluations. These provide stakeholders with a better understanding of a proposal's benefits and risks. As the fidelity of a prototype increases—as it gets closer to the final design—so do the costs of making changes. For this reason, we attempt to do as much development work as possible in a laboratory setting where we can uncover any critical issues or flaws before proceeding to a higher fidelity, and higher cost, environment. This approach reduces the cost of fielding new capabilities while also ensuring higher success rates and timely progress toward agreement.

The lab is also used to examine human factors issues. Before any change to an aviation system is made, the FAA must think about how workload, communication, safety, efficiency, and flexibility will be affected. Often these areas are best explored in an immersive environment. We design specific lab scenarios that systematically collect subjective and objective metrics to assess the strengths and weaknesses of the overall concept.

The ultimate goal of all these activities is consensus building. In the lab, pilots, controllers, traffic flow managers, dispatchers, and others can view the simulations together and discuss their concerns. We have found that this environment helps stakeholders to more rapidly reach agreement. Input that would take months to gather if each group of stakeholders were to test the concept separately can sometimes be provided within hours. This makes for a more efficient process, saving valuable time and resources.

For example, in the last two years we had representatives from 18 of the 20 U.S. Air Route Traffic Control Centers in the lab to evaluate high-altitude airspace redesign proposals. The controllers worked the system with the proposed airspace changes. We were able to quickly make changes to the system and scenarios to examine their various issues and suggestions, which helped facilitate consensus building among the controllers.

Inside the Lab

The integrated ATM Laboratory hosts an extensible, scalable, real-time distributed simulation environment. It was developed with a layered architecture that brings together an integrated set of simulation capabilities for human-in-the-loop experimentation, evaluation, and demonstration. Lab capabilities include:

- A medium-fidelity reconfigurable fixed-base cockpit simulator coupled to an out-the-window visualization system with a 124-degree field of view.
- An en route air traffic control simulation capability providing a sector suite configuration and interactive simulated pilot (sim-pilot) positions. The sector suite configuration encompasses emulated radar displays and strategic planning and decision-support tools, including a conflict probe and flight information management tool. The en route simulation capability can simulate any number of sectors across the country with current or future airspace adaptation.
- A Terminal Radar Approach Control simulation capable of simulating a variety of current and future terminal environments. This simulation capability also includes sim-pilot positions.
- A tower simulation capability that includes a Bright Radar

CAASD staff and air traffic controllers participate in a simulation.

Indicator Tower Equipment display and provides surface movement of aircraft.

- The Collaborative Routing Coordination Tool (CRCT), a prototype of an integrated collection of traffic management automation functions. These help traffic flow managers to develop strategies for alleviating congestion and avoiding severe weather by allowing them to analyze the impact of proposed strategies. With the CRCT capabilities, the traffic manager is also able to visualize the impact of these different traffic flow management strategies.
- Portable Aviation Visualization Environment, a 3-D visualization environment that is used in conjunction with the cockpit simulation/tower simulation or separately to provide visual representation of physical and abstract concepts.

Toward a More Efficient Future

As MITRE continues to support the FAA and international community in finding cost-effective ways to meet the increasing demands on global aviation, we will continue to evolve faster and more productive ways to explore future concepts, including:

- **Mobile Labs**: Taking our capabilities to the stakeholders for early concept exploration through the use of smaller computers and portable tools.
- **Virtual Simulation Environment**: Developing a simulation framework that leverages the resources and capabilities of government- and industry-funded laboratories and facilities to further explore proposed concepts through the integration of these resources through distributed simulation.
- **Web-Enabled Simulation**: Providing access to laboratory-developed capabilities to a larger number of global participants via Web- and Internet-enabled technologies.

Laboratory prototyping and human-in-the-loop experimentation have proven to be cost-effective tools for defining and refining future ATM concepts. The integrated ATM Laboratory has enabled us to provide our sponsors with critical insight into technical solutions to a number of timely issues, including runway safety, airspace redesign, and the application of the Automatic Dependent Surveillance-Broadcast/Cockpit Display of Traffic Information to procedural applications. (*See sidebar, which follows on page* 6.) Our laboratory capabilities play a crucial role in developing and evaluating these new concepts, building confidence among key stakeholders, and advancing new ideas to implementation in a timely fashion.

For more information, contact Urmila Hiremath, *urmila@mitre.org* or 703-883-7406, or Glenn Roberts, *groberts@mitre.org* or 703-883-6820.

Why the Maze on the Cover?

Mazes are considered puzzles that must be solved to reach the goal—which may be finding your way to the "treasure" in the middle or finding your way in one side and out the other. Often in using experimentation to find the solution to a complex problem, we feel that we are in a technical maze. We come up with an idea and head down a path. Sometimes the path leads straight to a solution or an important revelation (an aha!). Sometimes we discover it's the wrong path, and then we turn in a new direction—by hurdling technical barriers in our way, by collaborating with people who have additional information, or by going back to the starting point and taking a new direction that looks more promising.

Hands-on Testing in the Lab Moves Procedures Forward

By Randall Bone and Dave Domino

Before the FAA approves new procedures for air traffic control, aviation stakeholders must work together and agree that the changes will be safe and effective. An example of how the ATM Lab can support key decision-makers is the development and testing of the Cockpit Display of Traffic Information (CDTI) Assisted Visual Separation (CAVS) concept. (This concept was formerly called CDTI Enhanced Flight Rules or CEFR.) The ability of the lab to provide "hands-on" interaction with the CAVS procedures was a critical step in building confidence among stakeholders that the concept was a clearly feasible implementation of CDTI technology.

CAVS is designed as an extension of current visual approach and visual separation procedures. Visual separation requires the pilot of the following aircraft to sight the leading aircraft and then maintain a safe distance behind it during the rest of the approach and landing. In this procedure, the pilot assumes responsibility for separation from the aircraft ahead. (However, if a pilot loses sight of the airplane ahead, he or she must advise the controller who will then reassume responsibility for separation and may have to start using instrument approach procedures.) When air traffic controllers can use pilot-applied visual separation, the typical result is tighter spacing between landing aircraft than is possible when instrument approach procedures are used. In other words, more planes can land in a shorter amount of time when visual separation is used.

The purpose of CAVS is to maintain airport capacity by delaying the transition from visual approach operations to instrument approach operations as weather conditions deteriorate. The CDTI supplements out-the-window visual contact and allows the pilot to lose sight of the aircraft ahead while still keeping it "in sight" on a traffic display. By expanding the weather conditions under which visual separation may be applied, airport capacity may be maintained and delays reduced.

Despite its potential benefits, CAVS did not immediately win over all aviation stakeholders. Some air traffic controllers feared that CAVS was a move toward making pilots responsible for "self-separation" from all aircraft—a concept they did not find acceptable. Some people in the FAA standards community also expressed strong reservations about the use of CDTI for aircraft separation, including concerns about pilots accepting extra responsibility.

CAASD played a major role in moving the CAVS concept toward acceptance, principally by bringing pilots, controllers, and others into the lab to fly the concept and test its strengths and weaknesses. We were able to work with stakeholders in real time to resolve misunderstandings about CAVS.

It was important to make sure everyone witnessed and understood CAVS' Automatic Dependent Surveillance – Broadcast (ADS-B) and CDTI technologies and their applications. We briefed the CAVS concept to pilots, controllers, and various aviation organizations. We also had regular contact with United Parcel Service, which has been planning to implement CAVS. These efforts helped the aviation industry understand the concept's potential and reduced the level of skepticism.

The most powerful "myth-busting" tool, however, was the CAASD ATM Laboratory. Without the lab and its simulations, it would have been almost impossible to effectively communicate the CAVS concept and its benefits. We found that people who experienced the procedure by watching the simulations, or by actually flying the procedure in the ATM Lab cockpit (*see photos above*), had a much more positive view of CAVS than those who had only read about it. For example, one key FAA individual (also a pilot) who came to MITRE for a discussion of CAVS admitted to us that he had come in as a skeptic. After flying the CAVS simulation in our cockpit simulator, however, he left convinced that the procedure was worth further exploration.

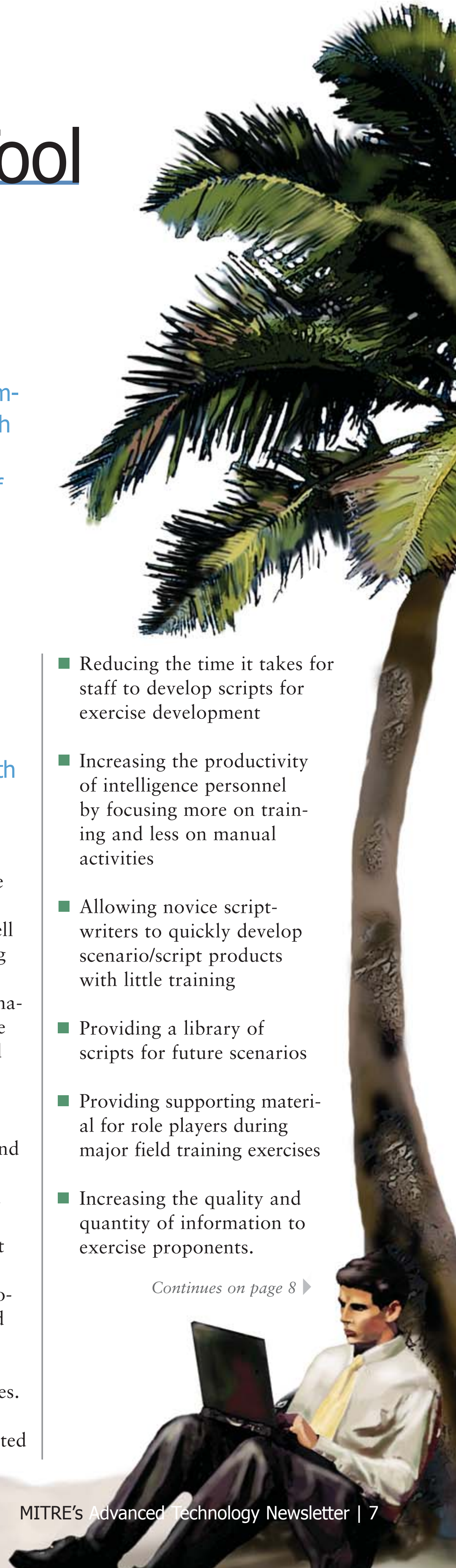
The lab also allowed us to gather feedback that would help us improve the CAVS concept. We evaluated the experiences of pilots and air traffic controllers who participated in a series of CAVS simulations and found that pilots were comfortable with the display features of the CDTI, were willing to accept responsibility for separation (as they do now with visual approach), and felt their resultant workload was acceptable.

As a result of this research, and from stakeholder feedback, we determined that the aggressive original proposal for using CAVS during instrument approach procedures should be redefined as a long-term goal. For the near term, we recommended the use of CAVS during visual conditions with marginal visibility (e.g., five miles), where visual contact with the aircraft ahead may be lost temporarily, but the trail aircraft is always in visual conditions. This application was judged by most stakeholders as a more acceptable first step. It was the ATM Lab and its capabilities that helped mold CAVS into an operationally feasible concept.

For more information, contact Randall Bone, *bone@mitre.org* or 703-883-5655, or David Domino, *ddomino@mitre.org* or 703-883-3695.

Rapid Prototype Process Creates State-of-the-Art Intelligence Training Tool

By Michael C. Higgins



Training military intelligence staff includes developing complicated scenarios and exercises that are realistic enough to sharpen critical skills. For example, in the scenario "Terrorist Attack on the Island of Oahu," several different kinds of attacks occurred within a short timeframe—involving bombings, riots, and chemical attacks—launched by two separate groups. Numerous military services from the United States, as well as from Australia, Canada, and the United Kingdom, participated in this 2003 exercise. The scenarios were richer and easier to produce, thanks to a tool called the Counterintelligence and Human Intelligence Exercise Scripting Support System (CHESSS). MITRE was instrumental in creating this capability through the use of a rapid prototyping approach and is transferring the technology both to the Department of Defense and to commercial manufacturers.

The need for a tool such as CHESSS was highlighted by the terrorist attacks of September 11, 2001. The U.S. Pacific Command Joint counterintelligence coordinating staff (USPACOM J2XCI) —which leads and coordinates joint and combined military counterintelligence, human intelligence, and criminal investigative service operations—has worked hard to improve its field training and command post training via the Vigilant Shield joint and combined exercises held in Hawaii and Australia. This series of exercises aids in the evaluation of the effectiveness of the command, control, communications, and computers used by the joint groups in support of combined military counterintelligence, human intelligence, and criminal investigation.

The USPACOM J2XCI asked MITRE to help design, develop, and deploy a new secure Web-based prototype to improve the richness and quality of the Vigilant Shield scenarios, as well as the efficiency of the scripting development process. The CHESSS portal provides information and tools to help configure exercises, create story lines and roles, assign locations, etc.

Before CHESSS, USPACOM J2XCI used ad hoc processes and manual methods to develop its complex exercise scenarios and scripts. It had few documented procedures available to support its novice scriptwriters. Thus it took a great deal of time to produce high-quality scenarios and scripts, particularly those that required collaboration among participants spanning time zones.

CHESSS, however, saves limited resources by:

- Reducing the time it takes for staff to develop scripts for exercise development
- Increasing the productivity of intelligence personnel by focusing more on training and less on manual activities
- Allowing novice scriptwriters to quickly develop scenario/script products with little training
- Providing a library of scripts for future scenarios
- Providing supporting material for role players during major field training exercises
- Increasing the quality and quantity of information to exercise proponents.

Continues on page 8

Continued from page 7

Rapid Prototype Process Creates State-of-the-Art Intelligence Training Tool

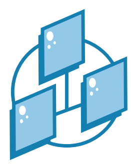
Lessons in Prototyping

The successful development of CHESSS illustrates important lessons in concept development, prototyping, and technology transfer.

Lesson 1: Share Information

Place an early and continuing emphasis on communications among all team members to build a close working relationship. This is crucial to understanding the sponsor's needs and concerns; and through regular communication you can clearly define a sponsor's problem and the roles and responsibilities of the various stakeholders involved in the project. Also, through communication you can help educate the sponsor on the various ways in which current and emerging technologies could be used in solving the problem. It's also important to remain objective and neutral while discussing requirements with the sponsor to ensure a clean development slate.

With CHESSS, we set up regular communications between the MITRE team and the sponsor's team, as well as among MITRE team members located in Bedford, Massachusetts, McLean, Virginia, and the sponsor's base of Honolulu, Hawaii. We employed all the communication methods available to us: video teleconferencing, teleconferencing, MITRE's internal Web site, a list server, e-mail, Instant Messaging, and face-to-face visits.



Lesson 2: System Design

Appraise all factors when determining an appropriate prototype design approach. Carefully consider the duration of the project; the availability, amount, and type of funds; development resources, including hardware, software, and labs; the number and type of sponsor agencies; the complexity of the proposed prototype; and security and accreditation requirements.

When MITRE weighed the factors involved in developing CHESSS, security was one of our chief concerns. Because CHESSS would operate at the Secret level, security drove the design, choice of tools and technologies and engineers (clearances), and created a requirement for accreditation.

Lesson 3: Storyboarding

Storyboarding helps ensure that the development team understands the requirements of the sponsor, and it enables the team to more clearly articulate the technical requirements to members not familiar with all aspects of the work. Storyboarding serves as an excellent visual tool. However, the first prototype applications often prove to be the best means by which sponsor and engineering teams can understand the evolution of the prototype and refine it further.

MITRE created an interactive storyboard, using MS Visual Studio Net, of the various CHESSS pages. This storyboard allowed users to understand the design layout, functions, links to databases, and other features.

Lesson 4: Spiral Development

Using a spiral development model is often conducive to prototyping and rapid application development. Spiral development promotes an iterative process between the sponsor and development team and allows for a continuous reevaluation of the system to refine its design. The phases of development (requirements analysis, design, development, and testing) are seen as a cycle in the model, and you can build up the functionality and quality of the product incrementally with each round of the cycle. Through this model, developers can work closely with stakeholders through all phases of development in setting and managing expectations. This helps the sponsor take ownership of the prototype and thus encourage its development, production, and implementation.

We employed a rapid prototyping model based on a timeline of days, not months, in developing

CHESSS. By using the MITRE Web portal to host the early prototype (keeping the project development unclassified as long as possible), we made it easier to demonstrate CHESSS to the sponsor and to allow stakeholders in locations across the globe to review, test, and provide feedback on the project. Through the feedback supplied by the spiral development process, our team compiled users' lists of "wants and needs" to ensure that all issues were considered, agreed on, and prioritized. We demonstrated the prototype to the sponsor on a weekly basis. Each cycle allowed MITRE and the sponsor to define "objectives for success" for each milestone.

Lesson 5: Training

When building a unique prototype, it is important to develop a hands-on training program to ensure that all users thoroughly understand how to use it. After each training session, gather feedback and recommendations from end users through discussions and surveys, and use this feedback in improving the system.

MITRE employed just such a training program to ensure that USPACOM J2XCI staff and coalition users (from Australia, Canada, and the United Kingdom) were sufficiently skilled in the application of CHESSS to use it successfully in the Vigilant Shield exercises.

Lesson 6: Testing and Documentation

Capture key installation and user documentation notes during development, rather than try to retrace those steps later. Testing and documentation help keep development teams focused on the fact they will eventually have to turn their prototype over to the sponsor, who will have to know how to install, use, and maintain the system.

Through MITRE's thorough documentation efforts, the USPACOM J2XCI had no difficulties in employing CHESSS when it debuted at the Vigilant exercise.

Lesson 7: Technology Transfer

When developing a prototype that may be used outside the sponsor's domain, make sure to use an adaptable design.

As CHESSS would not be limited in use solely to USPACOM J2XCI, we designed the prototype so that it could easily be adapted to domain areas beyond intelligence. CHESSS now supports exercises around the world, with a user base that includes the 8th U.S. Army in Korea, the Counterintelligence Field Activity, the U.S. Southern Command, the Joint Readiness Training Center, and the Joint Forces Command. We transitioned the prototype software system to the U.S. military and its contractor engineering support team so that the contractors could develop the operational system for further configuration control, enhancements, operation, and maintenance.

While delivering the CHESSS prototype to the U.S. military was a straightforward process, honoring the requests of foreign governments such as Australia, Canada, and the United Kingdom for copies of CHESSS required our team to look to the MITRE Technology Transfer Office for help.

MITRE began developing CHESSS in July 2002. Following the lessons described above allowed us to help put CHESSS into operation in time for the Vigilant Shield exercise held in July 2003. We continue to work closely with sponsors to transfer CHESSS to other military groups who will apply this technology in a wide variety of exercises.

For more information, contact
Michael Higgins, *higginsm@mitre.org*
or 703-883-3069.
A DVD is available that showcases the CHESSS case study.

Experimentation and Prototyping Laboratories Forge Military Process and Product Improvements

By Lewis A. Loren

Joint force experiments give the military a chance to test new systems and approaches and find out what works and what doesn't before these tools are used on the battlefield. One good place to practice is in the Air Force "SWIFT" Laboratory. The lab presents an ideal environment in which to use realistic simulations to test, observe, and practice both the science and art of successful joint military operations.

The Software Integration Facility Testbed (SWIFT) Lab was designed to support the evolution of Time-Critical Targeting Functionality (TCTF): that is, being able to detect, track, and destroy a time-sensitive enemy target (such as a mobile SCUD missile launcher) in the most efficient way possible. We helped create a series of simulations and experimentations (SIMEXes) to take advantage of the lab and all it offers. Sponsored by the Under Secretary of Defense for Acquisition, Technology, and Logistics, SIMEXes bring together participants from various locations to engage in war-game scenarios. Developers from the Air Force, Army, Navy, and other organizations use SIMEXes to test how well their TCTF systems interoperate—testing systems, functions, and human factors.

Two years ago, we used research and TCTF Program funds to build the SWIFT lab, which connects other MITRE and sponsor labs, so that we could help advance TCTF. In three years, we've conducted 14 SIMEXes and helped fine-tune such important new systems as Cursor on Target (*see sidebar*). The combination of our staff members' expertise in systems engineering and the participants' experience on the battlefield makes the lab a powerful tool. These exercise participants, including military personnel who manned time-critical targeting cells in Afghanistan and Iraq, try out new prototypes in the lab and give us feedback. They are the best people to compare new approaches with how things are done currently in the field.

For example, in the past (and in some cases, today) special ops warriors would find themselves on the battlefield trying to get information about a target back to their command post. They would use a handful of non-integrated machines and manually perform calculations, which they would call in during long voice transmissions over noisy radios. The room for error was large. There was no integrated electronic means to view data gathered in the field and to pass their analysis of that data along to the command chain of the various military branches. Last year, however, some of these warriors helped prototype a new automated Cursor on Target system that sends time-sensitive targeting information from machine to machine, reducing the chance of human error.

The most recent SIMEX was conducted in June and involved 145 distinct simulation and command and control systems manned by more than 30 operators. In one of the exercises, participants investigated the extent to which an Advanced Field Artillery Tactical Data System improves coordination of information. In another simulation, Navy personnel manned an Air Force weapon-target pairing system to determine whether offering the various military services systems that were developed by the other services could aid collaboration.

After each SIMEX is over, we analyze data on what worked, what didn't, and why. The end result is improved DOD tactics, techniques, procedures, and archi-

MITRE and Air Force staff network with labs in other locations during the 2004 joint simulation exercises.

tectures. For example, we are now helping to develop systems to deliver TCTF data electronically across integrated networks. These systems generally comprise two components: a weapon-target pairing system and an Automated Assistance with Intelligence Preparation of the Battle Space system.

The lab contains numerous technologies and DOD systems, as well as databases containing information about DOD assets, munitions, targets, and enemy installations to support the exercises and simulations. Just a few of the TCTF systems that have been tested during the SIMEXes include Cursor on Target; the Blue Force Tracking Initiative, which focuses on tracking allied forces to help avoid friendly fire casualties; and Work Flow Manager, a system developed to coordinate personnel activities across the services.

One added benefit to the SWIFT lab and the SIMEXes is the environment they impart—and not just in terms of technology. The laboratory provides a setting in which failure is allowed. Developers aren't putting their systems into competition; they're exploring the best ways to integrate them and enhance them. System failures during the simulations bring not black marks and bad press, but opportunities to learn, to seek solutions, and to pursue better alternatives.

This philosophy of honest appraisal frees SIMEX participants to focus on the crucial questions: Where did the information or system break down? Were these breakdowns due to operator procedures, incorrectly formatted data, or interoperability problems between systems? How can this be solved?

Of course, MITRE has many other labs throughout the company devoted to solving our sponsors' challenges in warfighting, aviation, and enterprise modernization. These facilities hone our staff's skills and enable us to research and validate emerging technologies and experiment with fielded systems or simulations to determine user impact and the appropriateness of advanced technology. What works and what doesn't? Better to find out before you go into the field.

For more information, contact Lewis A. Loren, lloren@mitre.org or 781-271-5969.

"Cursor on Target" Improves Efficiency

By Rich Byrne

Labs give us an environment to test new ideas, such as machine-to-machine automation that can make time-sensitive targeting three times faster and significantly more accurate. In a 30-day rapid prototyping effort, MITRE addressed a problem highlighted by the Chief of Staff of the Air Force, General John Jumper. He wanted to replace the old way of targeting— which involved time-consuming, error-prone human translations—with an automated way of passing key targeting information (what, where, when) across multiple machines.

In creating the new system, a MITRE team fused target information from a laser range finder, a compass, and a GPS receiver and then sent the data to an intelligence system to be refined for high-precision resolution. From there the data was relayed over a Link 16 radio to an F15E jet fighter to be automatically downloaded to onboard precision-guided munitions. All these interchanges were handled machine to machine. The idea is for battle commanders to be able to mouse over an aerial view of enemy positions, point, click a cursor, and watch as the target is eliminated. Cursor on Target provides real-time access to secure and reliable information.

We conducted some of the early tests of the Cursor prototype in the SWIFT lab, and later tests were conducted with live-fly exercises. The result was a series of rapid spirals that culminated in the deployment of Cursor on Target as part of the M2MT (Machine-to-Machine Targeting) system. This success has spawned more than 50 other Cursor on Target prototypes, a large number of lab demonstrations, and the approval for a number of additional rapid deployments.

For more information, contact Rich Byrne, rbyrne@mitre.org or 781-271-2811.

The Path from Concept to Reality: Getting Information to the Warfighter in the Field

By John Kane

Ideas for innovative technology often come from the challenges experienced by engineers in the course of their daily assignments.

These ideas, however, must be pushed along the road to development—with all its ups and downs—to make it into the field. That's what happened to the Intelligence, Surveillance, Reconnaissance (ISR) Information Service (ISRIS) concept conceived back in 2000 after the Commander of the U.S. European Command Joint Analysis Center (JAC) asked MITRE to support the NATO Linked Seas 2000 Exercise. Specifically, the JAC needed engineering help to link to the Global Hawk Unmanned Air Vehicle (UAV) ground station located near Norfolk, Virginia, so it could receive, exploit, and disseminate Global Hawk imagery collected over Portugal during the exercise.

Although the event was successful overall, two problems presented themselves: the lack of mission situational awareness information, and stovepiped access to sensor data. MITRE started looking for solutions—that is, a better way for warfighters to discover and access near-real-time data from airborne surveillance platforms using Web-enabled tools.

A MITRE team began working on the internal R&D effort called ISRIS, which was designed to push Internet technology as close to the ISR sensor platform ground station as possible and to use that technology to provide warfighters with access to near-real-time ISR data via the operational battlefield networks. We developed an ISRIS prototype server that demonstrated to our sponsor how warfighters, using just a browser, could monitor Global Hawk and Predator UAV missions in an interactive map display and gain near-real-time access to sensor data as it was collected.

The Road Widens

Our work fit into the vision outlined in 2001 by John Stenbit, then the Assistant Secretary of Defense for Command, Control, Computers and Intelligence. He promoted his goal of augmenting the traditional "task-process-exploit-disseminate" approach with one that would "post before process" all data to the emerging Department of Defense Global Information Grid (GIG). [The GIG is a globally interconnected, end-to-end set of information capabilities, processes, and personnel for collecting, processing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel.] This new process was termed "TPPU" or "task, post, process, and use," and was designed to help users make faster, more informed decisions in the field.

MITRE saw how ISRIS could realize Stenbit's objective and presented the concept and prototype to him. His response was: "MITRE's ISRIS prototype really fits my vision of the future, where ISR information will be made available, shortly after it is collected, to the warfighters who need it the most. It's a great demonstration of what I call 'post before process' for populating the network with new, dynamic sources of information needed to defeat the enemy."

In July 2002, MITRE published a paper on ISRIS, explaining the concept and lessons learned, which was highlighted in the Jan/Feb 2003 volume of the *ISR Journal*. The Air Force used the paper to help defense contractors bidding on the Distributed Common Ground Station (DCGS) 10.2 contract understand the concept of task-post-process-use as it related to the ISR community.

The next step in the prototype's development occurred when the U.S. Joint Forces Command (USJFCOM) decided to invest in the ISRIS concept and push it toward operational use. In 2003, the Joint Staff approved the seed funds to further develop the prototype. Most new ideas in the Department of Defense (DOD) go through a competition for funds and resources so that only the best ideas move forward. USJFCOM tasked MITRE to lead ISRIS through the FY04 Advanced Concept Technology Demonstration (ACTD) Program of the Deputy Under Secretary of Defense for Advanced Systems and Concepts.

We then came to an intersection in the road to concept realization. The Air Force Electronic Systems Center (ESC) presented a coalition-focused concept to USJFCOM that complemented MITRE's ISRIS concept. USJFCOM approved our suggestion to combine the two ideas—

creating a stronger concept proposal—and the Multi-sensor Aerospace-ground Joint ISR Interoperability Coalition (MAJIIC) was born. MAJIIC was selected as a new start 2004 ACTD, and we also received support from the Air Force and Army. This was a big push forward on the road from good idea, to MITRE demonstration of capability and feasibility, to implementation.

Because of our subject matter expertise, the USJFCOM appointed MITRE as the MAJIIC ACTD Technical Manager. In this role, we are helping to evolve the original ISRIS concept, employing MITRE's wide range of technical capabilities to meet our sponsor's mission needs.

Net-centric Technology

MAJIIC's objective is to enhance U.S. Joint and Coalition ISR interoperability via the development, testing, and implementation of data standards, XML schemas, common programming interfaces, and leading- edge Web-based enterprise services. The ACTD endeavor consists of two tightly coupled efforts: The Multinational MAJIIC Project and the MAJIIC Horizontal Fusion Initiative. The MAJIIC Project is a collaboration among the United States and eight nations focused on the development, testing, and implementation of the technology and processes necessary to obtain seamless ISR data sharing in a coalition environment. The MAJIIC Initiative is part of the DOD's Horizontal Fusion Portfolio, an effort to develop, demonstrate, and field the first GIG Enterprise Services Service-Oriented Architecture on the operational Secret Internet Protocol Network. MAJIIC links ISR sensor platforms into the GIG via leading-edge Web services.

We are heading the development of two ISRIS server configurations that support MAJIIC: ISRIS Multi-INT serves imagery, moving target indicator, and electronic intelligence sensor data from manned and unmanned airborne surveillance platforms. ISRIS Video serves video sensor data from UAV platforms. Both servers, which handle platform mission data as well, will be co-located with the ISR platform ground station or data downlink location in small-footprint deployable configurations for use in the field. This distributed architecture enables the ISR platform's data stream to be posted directly to the GIG, where the information can be accessed by all users.

ISRIS automatically pre-processes ISR sensor and mission data to ensure that posted data is compliant with DOD Data Discovery Metadata Standards, security labeling standards, XML taxonomy standards, and data format standards. Supported standards (all critical for interoperability) will vary depending on the operating environment and will evolve as the GIG and coalition environments mature.

One of our primary focuses was to provide support for low-bandwidth environments with solutions for imagery and video dissemination. Users can access ISRIS services via standard DOD-approved Web browsers. ISRIS will also service requests from network-based applications and value-adding systems that conform to the GIG, Coalition, and DCGS interface specifications and security policies.

The Last Mile

MAJIIC's ISRIS servers were demonstrated publicly for the first time at the Quantum Leap 2 demonstration in August 2004. The ISRIS Multi-INT server received live ISR data from an in-flight Joint STARS platform and provided near-real-time task-post-process-use access for the warfighters. The ISRIS video server disseminated simulated Predator video to warfighters using a standard Web browser. MAJIIC is scheduled to be operational in 2005 in support of the XVIII Airborne Corps' deployment to Iraq. MAJIIC will also be transitioned into future releases of the DCGS 10.2 Program of record.

It took four years, and lots of energy, to go from concept to reality, and prototyping played a significant role along the way. We had the expertise and resources necessary to move forward, with sponsors removing barriers from the path. As a result, warfighters will soon have unfettered net-centric access to near-real-time ISR information in support of their time-critical missions.

For more information, contact
John Kane, *jkane@mitre.org*
or 757-673-5706.

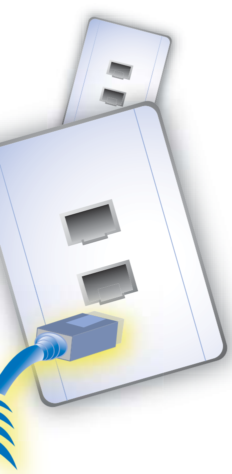
Creating a Plug and Play Server Prototype Fills Technology Void

By William Wollman, Harry Jegers, Beth Loftus, and Caleb Wan

As technology advances to the front line, it is putting increased capabilities into the hands of ordinary soldiers. Armed with wireless networks, powerful lightweight laptop servers, and handheld computers, a soldier can upload tangents for accurate bombing runs or download the latest satellite photos pinpointing the location of enemy units lying in wait.

One drawback to this "network-centric warfare" technology is maintaining connections on the battlefield. Soldiers are moving continuously, and this rapid movement in current military operations has created challenges. When they set up computers on the battlefield, how do they find servers to connect to? What do they do if the server to which soldiers are connected goes down or if a shift in the battle line forces the soldiers to move out of range of that server? Reconnecting to the network can require the soldiers to know how to reconfigure the system or a network router. Off the battlefield, a network administrator would be called in to perform these tasks. The military cannot afford to train all its soldiers in network administration.

A MITRE team began looking at state-of-the art solutions to this problem. What if the soldier's computer and the network with which it was attempting to connect could automatically configure themselves? What if soldiers' computers were just like their radios or walkie-talkies: all they had to do was turn them on and they would work, even if the soldiers moved out of range of their servers or if their servers were knocked out of the networking system?

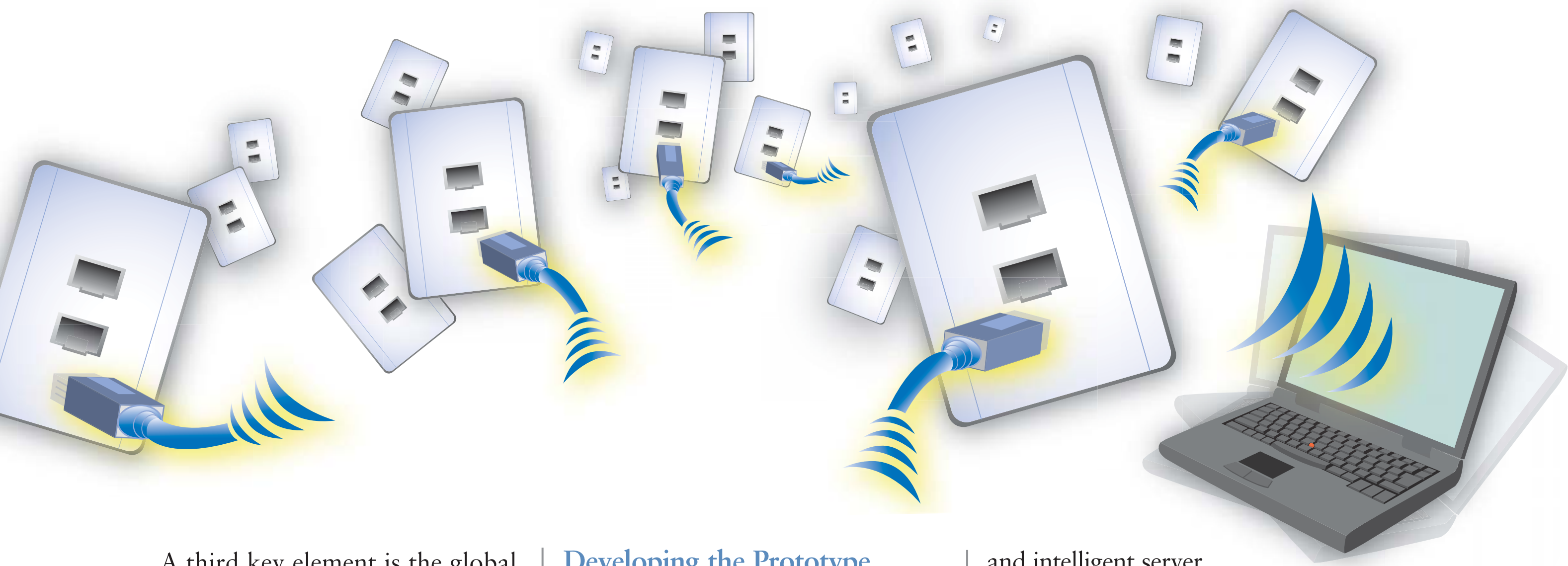


The team focused on a capability called "plug and play" that would allow soldiers to easily and securely attach to a network to share communications anywhere and any time. Using plug and play technology, servers and clients can be enabled to securely affiliate with any command post's local wireless network access point with minimum planning. Once affiliated, basic network services can be quickly located by clients. The team saw this as a different way of looking at the connectivity problem, one that could benefit not just the military but other organizations looking for disaster recovery help.

To thoroughly assess networking challenges, a prototype was created. We worked with a combination of open source and commercial router systems to demonstrate the benefits and feasibility of protocols. Through prototyping, MITRE is able to assess capabilities, further basic technology, or demonstrate something new to fill voids in the commercial market. This prototype, which has been demonstrated successfully, employs anycast, server load balancing, and global server load balancing. Our work demonstrated a solution that can increase our military's capability to quickly and easily deploy resilient network architectures. This work could be transitioned to any tactical command and control environment.

The MITRE research team first looked into the technology available for this project. Options for data delivery included broadcast, multicast, unicast, and anycast. We chose anycast, which permits an address to be shared by a set of computers. Data sent to the anycast group will be delivered to at least one computer within the common set. Usually, the information being transmitted to an anycast address will be delivered to the closest anycast group member. The anycast address can be either a unicast network address or an application-specific address, such as a URL. Anycast techniques are being used by the Army and other organizations to solve challenges such as disaster recovery.

Another key element is the server load balancer (SLB), which increases network resiliency by permitting multiple servers to be represented as one destination. Requests received by the SLB will be delivered to at least one of the computers within its server farm. The SLB can also be configured to monitor the "health" of each individual server and automatically remove any particular server that goes down. (A single server failure or server maintenance time period will not affect service availability.) An SLB also enhances network security by monitoring and controlling server access. Throughout the Internet today, SLB technology is used to support many server deployments.



A third key element is the global server load balancer (GSLB). This enhances SLB by allowing multiple server farms to be separated and physically distributed throughout a network. Network survivability is enhanced through location diversity. The server farms maintain commonality by using a common domain name (e.g., www.anycast.org). By collecting and maintaining information about the health of the various server farms and network performance parameters, the GSLB can direct a client to a preferred server farm.

Our work focused on deployment simplification of a resilient network architecture that uses anycast techniques created with SLB and GSLB. As part of this, we then developed another piece of the puzzle: an SLB Registration Protocol (SLBRP). The SLBRP automates the anycast process to permit servers to locally register their services with an SLB and to allow users to connect to a server without manually configuring a network or complex server networking. Once a server is registered, the SLB monitors the health of the service and provides continued access. When a server detaches from the network, the SLB automatically removes the server information from its configuration. Meanwhile, the GSLB works with the SLB to direct users to the "best" server. Agents measure server response time and network performance parameters and provide this information to the GSLBs. When a server farm is removed from service, the SLB will de-register the farm and associated domain from the GSLB.

Developing the Prototype

The MITRE team considered building a detailed simulation to demonstrate to the military how our plug and play system would operate. We decided it would be more effective, however, to build a prototype of the SLBRP and the plug and play anycast environment. With a prototype we could demonstrate that such a system was possible, not only for a proprietary commercial system, but also for an open source system.

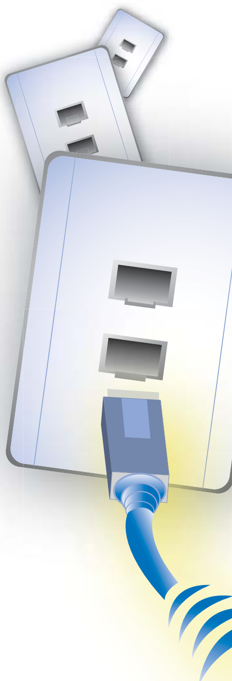
When considering commercial systems for prototyping, we chose an experimental version of the Cisco IOS that supports Java with an embedded Java Virtual Machine. This version allowed us to enable plug and play SLB networking with Cisco routers. We also leveraged the Cisco Service Assurance Agent technology for our network performance monitoring agents. In addition, we used the Linux Virtual Server and Zebra routing open source software for the project.

The prototyping process successfully demonstrated the benefits and feasibility of the SLBRP used with a combination of open-source and commercial router systems. We have demonstrated to the Army and Navy how the prototype provides mobile server support for anycast, plug and play server load balancing, and intelligent server selection via dynamic global server load balancing.

The SLBRP is a simple and efficient way to automate anycasting addressing. By leveraging plug and play SLB configuration concepts, we significantly helped to reduce the complexity associated with both server deployment and mobility in the battlefield. The concepts can also be used to enhance network configuration management within any organization.

Our work in this area points the way for the evolution of network-centric warfare. In the future, soldiers without any specialized computer training will be able to bring the full force of the military's information system to bear at every point on the battlefield.

Research and development work, including prototyping, allows MITRE to maintain our technical excellence while providing capabilities to the industrial community to advance particular technologies (in this case, networking). Our sponsors benefit through having more capable commercial products in the long run and through MITRE's continuing ability to guide industry toward capabilities that will solve sponsors' critical problems.



For more information, contact William Wollman, *wwollman@mitre.org* or 732-578-6475; Harry Jegers, *jegers@mitre.org* or 732-578-6383; Beth Loftus, *mloftus@mitre.org* or 732-578-6402; Caleb Wan, *calebwan@mitre.org* or 732-578-6468.

A New Approach to Analyzing Enterprise Architectures

By Paul Barr, Thomas J. Pawlowski III, and Steven J. Ring

In times of crisis—whether a terrorist threat, an earthquake, or a battle—it is critical for organizations to share information quickly and securely. How effectively information is passed from the source to the user can often determine the outcome of the emergency or conflict. To help ensure the quick and efficient flow of information, most U.S. government organizations—including the Department of Defense (DOD) and Department of Homeland Security (DHS)—have created enterprise architectures, which must be constantly tested and refined.

The idea of extending prototyping and experimentation approaches to architecture development is new and evolving. MITRE's goal is to help organizations analyze the information flow of complex systems-of-systems elements at the enterprise level, with the expectation that more efficient designs can then be defined at the software and hardware architecture levels.

All government organizations must create enterprise architectures to understand their IT needs and support investment decisions. These architectures act as a kind of roadmap for the design, development, and acquisition of complex, mission-oriented information systems. Such an architecture describes all aspects of an organization: its mission, organizational structure, business processes, information needs, software applications, and underlying technical infrastructure. A change in one dimension may impact the other dimensions. And an enterprise architecture is a living document—changing along with missions, strategic goals, and emerging technologies.

MITRE has a great deal of experience in creating enterprise architectures for numerous organizations, including the DOD and several of the DHS agencies. For example, we helped develop the DOD Architecture Framework, which all DOD organizations use as a basis for building and describing their own architectures so that they can be compared and related across organizational boundaries. The idea is that using the same framework to develop systems will help them achieve interoperability.

Most DOD Architecture Framework products currently being developed provide "static" information (e.g., name, description, quantity) for operational, system, and technical views of an information system. The DOD Architecture Framework, however, also supports descriptions of dynamic or behavioral information in the form of models and descriptions of events. But there are no processes and tools to capture this dynamic information—tools that can validate an architecture and perform "what if" analyses. The tools available don't provide an adequate model for conducting detailed time-dependent behavior analysis of complex, dynamic operations and human and system resource interactions, which can't be captured with static operational models.

In response, MITRE began a project to:

- Develop a methodology to convert static architecture products into executable architectures
- Develop a federation of simulations that represent the mission threads (business processes), communications networks, and operational environment for the system being used

The result of our work was the Executable Architecture Methodology for Analysis (EAMA). An executable architecture is a dynamic model of sequenced events performed by humans and systems, for example, how troops share and access information on a battlefield. Some architectures provide only a "static" view of the systems they are simulating. A "static" view displays a system's capabilities but does not display how information is actually produced and used in real-time operations. EAMA is designed to provide a "dynamic" view. It displays how information is produced and used, by whom, and what resources they use in doing so.

EAMA was developed to perform a dynamic analysis of the DOD's command, control, communica-

"To achieve this simulation of system relationships, we incorporated within EAMA a federation of simulations that represent the three key components related to mission objective."

tions, computers, intelligence, surveillance, and reconnaissance (C4ISR) information architecture. Recently we have also extended our EAMA work through an internal research project focusing on how multiple agencies respond to a crisis.

Dynamic analysis enables a user to assess the impact of change and determine measures of performance and effectiveness. To perform a dynamic analysis, we first chose a measurable objective. Next, we chose data with which to measure the objective. We fed the data into the simulation and measured the objective. Based on the results, the simulation was modified and rerun until the objective was met. In this case, based on our analysis of a simulation run, we recognized the bottlenecks of the information flow among the staff members in a battlefield environment. Performance objectives were achieved by altering the staff relationships and rerunning the executable architecture. By measuring multiple objectives, we can assess an overall operation under different conditions. To offer a complete analysis, dynamic models must simulate the relations among multiple systems as information is networked from source to user in crisis conditions.

Federation of Simulations

To achieve this simulation of system relationships, we incorporated within EAMA a federation of simulations that represent the three key components related to the mission objective. The first simulation component is the combat simulation model that provides the operational environment in the context of the flow of the battle. The second simulation component, the process model, consists of two views: an organizational view of human and material resources and a process view of the flow of information through these resources. The third simulation component models the communications network as it channeled the flow of information.

How do EAMA's simulations perform? Key events in the combat simulation trigger business processes in the process model. As the process model plays out, it requests from the communications network model the amount of time it would take for information to pass from various source nodes to user nodes through the network.

Homeland Security

Working with MITRE's Ken Hoffman, we extended the EAMA work through an internal research project designed to examine the application of executable architectures to a case where multiple agencies were involved in a common mission. For the case study, we used a homeland security scenario called "Vigilant Sentry" that involved a mass migration from a Caribbean island into the southern United States. This scenario involved agencies in DHS, with the Coast Guard in the lead, and agencies in the DOD, State Department, and Justice Department, as well as state and local authorities.

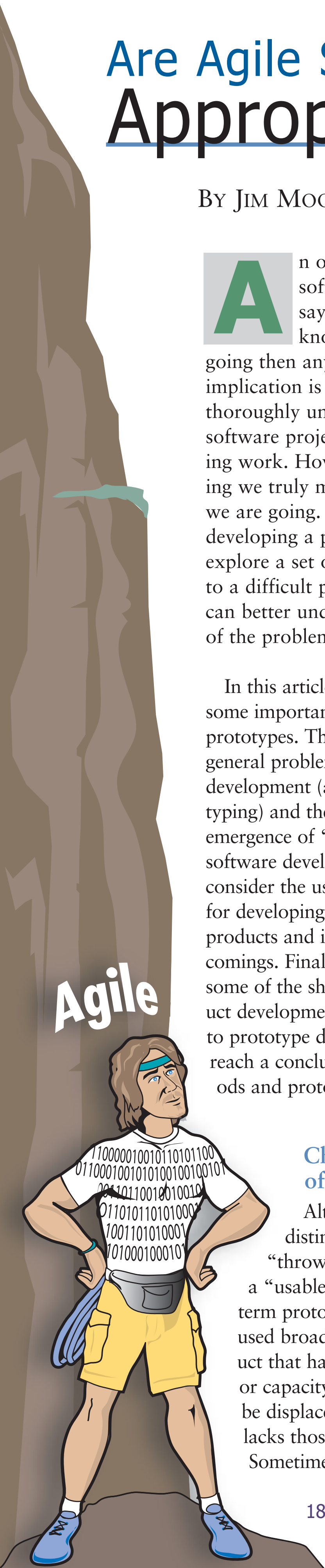
We narrowed the scope of the study to concentrate on the interdiction of migrant vessels by Coast Guard vessels and the reporting of the migrant information (name, gender, age, nationality, etc.) through command and control channels. Using the Vigilant Sentry operational plan and the architectures of several key agencies, we developed the scenario simulation using a Course of Action analysis tool called Joint Military Art of Command Environment (JMACE). We developed the key business processes in a process modeling tool called Bonapart. We also used a communications network modeling tool called NS-2 to represent the communications nodes involved in passing information among aircraft, vessels, and command and control centers. These tools were linked together and run as a federation of simulations.

An example of the sequence of actions among the simulations follows. In JMACE, Coast Guard aircraft and cutters patrolled their assigned areas in the Atlantic Ocean, Straits of Florida, and Gulf of Mexico. When one of these patrols spotted a migrant vessel in the water, it would send a detection report. This event in JMACE would trigger the business process in Bonapart, causing the activities within that process to begin executing. Typically these were information processing and decision-making processes at the command centers,

Continues on page 20

Are Agile Software Methods
Appropriate for Prototyping?

By Jim Moore and Tim Rice



An old proverb about software development says, "If you don't know where you're going then any road will do." The implication is that you have to thoroughly understand and plan a software project before commencing work. However, in prototyping we truly may *not* know where we are going. The purpose of developing a prototype may be to explore a set of possible solutions to a difficult problem so that we can better understand the nature of the problem.

In this article, we first look at some important characteristics of prototypes. Then we discuss the general problems of software development (a big part of prototyping) and the reason for the emergence of "agile" methods for software development. We next consider the use of agile methods for developing complex software products and identify some shortcomings. Finally, we observe that some of the shortcomings for product development are not relevant to prototype development and reach a conclusion on agile methods and prototyping.

Characteristics of Prototypes

Although some sources distinguish between a "throw-away" mockup and a "usable" prototype, the term prototype is generally used broadly to mean a product that has restricted function or capacity and is intended to be displaced by a product that lacks those restrictions. Sometimes the prototype is grown into the final product and sometimes it is simply thrown away and replaced. We sometimes develop prototypes to learn lessons because we're not sure what we want in the final product or we're not sure if the problem can be solved.

Complex modern products, and thus their prototypes, usually include software. In addition, prototypes sometimes use software in the place of components ultimately intended for implementation in hardware. This is done on the assumption that software components can be modified quickly in response to trial usage and experimental results.

Agile Methods

One of the challenging aspects of software engineering is the difficulty of constructing, within a limited schedule and budget, satisfactory products that meet the stated requirements, satisfy the implicit expectations of the users, and exhibit high quality. Because it is difficult to measure software product quality, the emphasis over the past three decades has been on defining high quality software development processes on the presumption that high quality processes will produce high quality products. As software systems became larger and more complex, so did the development processes meant to ensure quality.

The result, according to some software developers, is that the complexity of these processes has resulted in development that is slow, clumsy, and difficult—and has stifled creativity and supported mediocrity. These developers claim that when using these processes, they cannot rapidly and economically react to changes in system requirements that are simply a fact of life as users better understand their needs.

A group of developers started looking for a better way—faster and more flexible and responsive—to develop software. These approaches are now referred to as "agile methods." Generally, these methods minimize the overhead of development, including documentation, up-front planning, and controls on development. Instead, they emphasize frequent delivery of improved releases of increasingly capable software that can be directly evaluated by users of the system. By minimizing the overhead, they claim to make the software easy to change, hence reactive to changing needs.

Agile methods are a hot topic in commercial information technology because the proponents of the methods say they are faster, cheaper, and more responsive to user needs than the traditional methods of software engineering. A dozen or so different agile methods are becoming well known, with names like Extreme Programming, Scrum, and Internet Speed Development.

Agile methods generally have the following characteristics:

- The work flow is iterative rather than sequential. Developers quickly create an initial solution and then refine and improve it to approach the final solution rather than spending a lot of time and effort to get it right the first time.

- The software is the complete product. Developers don't write documents describing the requirements or the design of the product; instead, the software code speaks for itself.

- The development project is agile, i.e., in a short cycle the users try the current solution and tell the developers what they don't like and the developers change it.

Complex Software

Are agile methods appropriate in developing prototypes? We are still gaining experience and knowledge in this area. However, we do know that some of the characteristics listed above may be problematic for the development of large, complex, long-lived software systems. For example, the quickly developed code may "speak" to the original development team, but the lack of supplementary documentation may present a difficulty to maintainers years hence who are trying to figure out how to fix a problem or improve the product. The methods are particularly questionable for the development of systems with safety-critical needs. For example, we may not find users willing to try out partial solutions in the development of anti-lock braking systems. Safety-critical systems typically undergo many different forms of verification before users are allowed to try them.

Advocates of agile methods versus traditional software engineering methods generally divide along this question: Is a software product/prototype easy to change or hard to change? Traditional developers answer "hard" because they know they have to revise detailed development plans, modify the software, perform various reviews to ensure quality, and prepare extensive documentation explaining what they did. Agile developers, however, are more likely to say that changing software is easy because they don't have a detailed plan, don't perform reviews, and don't prepare much documentation. So to change the product, they just alter the code, test the changes, and try it out on sample users. If it's a disaster, they back it out. If it's not right, they fix it. It is the treatment of planning that provides the sharpest distinction between traditional methods and agile methods.

Suitability of Agile Methods

Regardless of their applicability to the development of large, complex software systems, agile methods do appeal to prototype developers. For one thing, the agile methods' focus on producing working software as the end product fits well with the needs of some prototype developers. Also, the quick turnaround of user evaluation—receive change request, implement code, have users re-evaluate—supports the use of prototypes to evaluate a variety of solutions to the users' problem.

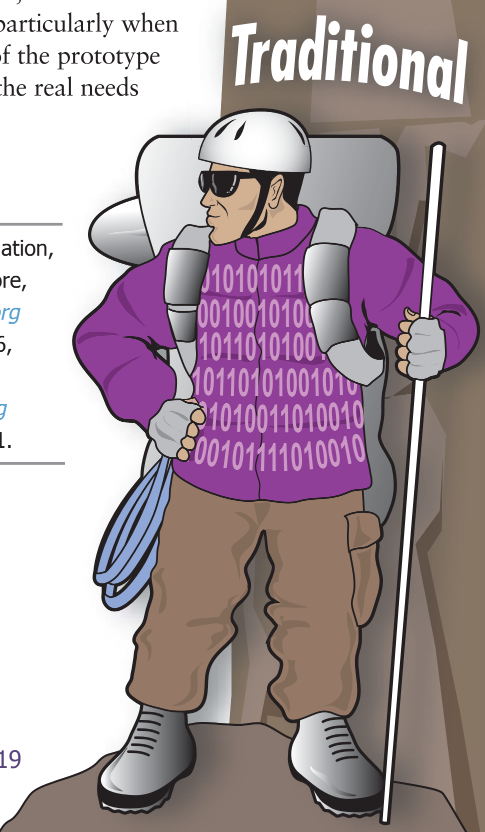
Some people may object that these methods are simply undisciplined "code and fix" programming without any underlying rigor. There is evidence, however, that some agile methods introduce just enough rigor to be suitable for prototyping. For example, the method known as Extreme Programming (XP) requires that programmers work in pairs—two programmers sharing one computer—so that every line of code will be read by two sets of eyes. XP also ensures comprehensive, but low-cost, regression testing by requiring programmers to write the tests for their code before writing the code and by automatically running those tests periodically on the evolving product to ensure that no function is ever lost.

Here's another way to look at the contrasts between the traditional and agile approaches to software development: There are two different ways to climb a mountain. You can climb slowly, following a detailed plan that anticipates every problem, leading a train of porters carrying all of the equipment that you might need to set up a string of camps. Or, you can climb fast and light, carrying only the essentials on your back, improvising as you go—hoping to get to the top before problems arise. Both ways can succeed; both ways can fail.

In building a prototype, we are often unable to make a detailed plan. The purpose of developing a prototype may be to explore a set of possible solutions so that we can better understand the nature of the problem. In this case, the heavy approach may not work because we don't know which problems to anticipate.

In short, the jury is still out on the question of using agile methods for building large, complex software products. The application of agile methods to prototyping, however, does seem like a sensible fit, particularly when the objective of the prototype is to discover the real needs of the user.

For more information, contact Jim Moore, *moorej@mitre.org* or 703-883-7396, or Tim Rice, *tbrice@mitre.org* or 781-271-4501.



***The EDGE* MP 04W0000122-02** *The EDGE* is published quarterly by The MITRE Corporation. www.mitre.org/edge

Publishers
Catherine Howland,
Stephen Huffman, and
Mark Maybury

Staff
Beverly Wood, Editor-in-Chief
Regina Hansen, Managing Editor
Christopher Lockheardt,
Lead Editor
John Schleith, Art Director
Maroe Susti (cover),
Mickey Stinnett, John Ursino,
Illustrators

Please send feedback to:
Regina Hansen
Telephone 703-983-7301
Email: hansen@mitre.org

Continued from page 17

A New Approach to Analyzing Enterprise Architectures

including, for example, determining which Coast Guard vessel would interdict the migrant vessel. As one activity was finished and passed information to another activity, Bonapart would request a delay time from NS-2. This delay time was the time required to send the information from the sending node to the receiving node based on the current load on the network. NS-2 would calculate this delay time and send it back to Bonapart, which would add it to the process time. When Bonapart completed the business process, an order would typically be sent back to JMACE instructing the appropriate U.S. vessel to interdict the migrant vessel.

By running this federation of simulations, we were able to examine several different measures of effectiveness and performance. From JMACE, we could determine the number of migrant vessels interdicted and the number that were able to reach the United States. From Bonapart, we determined the processing time required to complete the various business processes. We also were able to determine the use of the available resources that were needed to execute the business processes. If one of these resources had become a bottleneck in the system, we would have seen the effect on processing times. From NS-2, we could determine if there were any bottlenecks in the communications network that resulted in significant delays in passing data through the network.

The main focus of the case study was to extract insights and lessons learned about multi-agency executable architectures. We determined that multiple agencies working to achieve a common mission have many challenges, including the fact that many of their architectures lack the maturity and breadth to make them easy to integrate.

But we have shown that by linking simulation components together, we can create an executable architecture with which to conduct dynamic analyses. We continue to enhance EAMA, for example, by adding functions to refine the resulting analysis. These functions include dynamic handling of stale messages and adding context-sensitive selection of resources. We are also addressing technical issues, such as EAMA's current inability to handle fault tolerance. The goal is for EAMA to gain the ability to examine how troops, or other groups, and their supporting processes consume resources over time.

Our work provides the DOD and other organizations with the capability to better develop and analyze their architectures, support simulation-based acquisition, and support DOD and federal architecture communities. For example, it should be possible to improve descriptions of business processes, the activities behind those processes, and the information flow among the processes in enterprise architectures. The approach is consistent with the concepts of spiral development—which includes incremental iterations of planning, analysis, construction, testing, and release stages. A simulation-based architecture, or executable architecture, enables the creation of simulation-based or executable specifications for both existing and proposed systems.

We have provided this methodology to the DOD community via demonstrations, conference presentations, and papers. This work could be useful to many of MITRE's other sponsors who are developing their own architectures and related products. We believe that EAMA can be an important component of an overall architecture-based strategy in which investment decisions are directly linked to mission objectives and their outcomes.

For more information, contact Paul Barr, poobarr@mitre.org or 732-578-6307; Thomas J. Pawlowski III, pawlowst@mitre.org or 913-684-9139; Steven J. Ring, sring@mitre.org, 781-271-8613.

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102-7508