# ARTEMIS: A High-Fidelity End-to-End TBMD Federation

*Ann F. Pollack and Andreas K. Chrysostomou*

As the Mission Technical Direction Agent for the Navy Theater Wide (NTW) program of Ballistic Missile Defense, APL performs overall systems engineering analysis for design reviews, trade studies, and other technical forums. To support these analyses, the Laboratory developed the APL Area/Theater Engagement Missile/Ship Simulation (ARTEMIS), a distributed high-level architecture federation built on existing high-fidelity simulations of the NTW system components. ARTEMIS captures the crucial closed loop interactions between system components, providing a systems engineering tool for functions including performance assessment, design verification, and flight analysis.

## INTRODUCTION

Ballistic missiles have proliferated throughout the world and pose a significant threat to the United States and its allies. The Navy Theater Wide (NTW) Program is designed to defeat medium- and long-range ballistic missiles, providing a flexible and autonomous defense capability that can rapidly deploy throughout the world. APL, as the NTW Mission Technical Direction Agent (TDA), has an active role in design and development. TDA responsibilities include independent evaluation of system design concepts, assessment of the integrated performance of system elements, and analysis of high-risk areas to validate technical feasibility.

To support evaluation and assessment, APL has a suite of simulations covering the ballistic missile domain and representing multiple layers of fidelity. System-level simulations examine the entire battlespace, providing a general picture of performance ranges and defining the boundaries of operation. The APL Defended Area Model is an example of an NTW system-level simulation. Closer examination of particular scenarios or critical systems requires high-fidelity simulation where actual physical details and precise algorithms are represented. The Laboratory has detailed physics-based simulations of the AN/SPY-1 radar (FirmTrack), Standard Missile-2 (SM) guidance (SM-2 Block IVA and SM-3 six-degree-of-freedom [6-DOF] simulations), and SM signal processor (Ballistic Missile Localization and Selection-Tool [BLAST]). These simulations have supported numerous studies for NTW and its predecessor program, the Aegis LEAP (Lightweight Exo-atmospheric Projectile) Interceptor (ALI).

## MOTIVATION

Often in these studies, in addition to evaluating the individual components, APL is called upon to provide the overall systems engineering perspective. Through its role as TDA and as a trusted agent of the government, the Laboratory has expertise across all NTW elements—the threat, ship systems, and missile systems (Fig. 1). Our suite of high-fidelity simulations provides detailed analysis of these elements. In the past, results have been fed from one simulation to the next to yield an overall system assessment. Although this level of assessment has been very useful, it cannot capture all of the dynamic interactions that occur among elements. It can also result in prohibitively large data exchange files. However, an end-to-end integrated system simulation can capture dynamic effects and perform rapid data exchange enabling higher-fidelity, more powerful systems engineering.

Certain aspects of an engagement could theoretically be modeled without an integrated end-to-end simulation. For example, it is possible to build a time-dependent, iteration-dependent radar track file to provide target data to the missile. The issue becomes the amount of required data. These simulations use Monte Carlo analysis, executing multiple iterations where certain elements vary randomly and capturing the inherent uncertainties in the problem domain. But again, files representing all of the required information from 100 or more iterations become prohibitively large. Thus, modelers fall back to statistical summaries instead of the higher-fidelity iteration-by-iteration variations. An end-to-end simulation can exchange these data during the individual iterations, eliminating the need for static storage and exchange of huge blocks of data.

More efficient data exchange is clearly an advantage of end-to-end simulation; however, the biggest payoff in terms of fidelity comes from the ability to model the dynamic effects of the system components on each other. A simple example is the fact that the radar tracks the missile during its flight, impacting the resources available for other radar functions. In a more complex example, the radar continues to track the threat complex during missile flyout. If the radar determines that a different object within that threat complex should be targeted, it can communicate that discovery to the ship's onboard Weapon Control System (WCS), which can uplink new guidance commands to the missile.

As another case in point, a particularly tight coupling exists between the missile guidance and the missile signal processor. Guidance points the signal processor, which in turn selects a target, then tells the guidance where to point and fly. These are both complex systems represented by their own stand-alone simulations. They become a much more powerful analysis tool when the loop between them is closed. Not only can one identify if the missile picked the correct target object, but also whether that decision was made in time to divert and intercept the threat.

Many more examples exist of behaviors that are impossible to model without an end-to-end simulation. Through its systems engineering efforts, APL recognized the value of capturing these complex interactions and began development of an end-to-end NTW system simulation.

At about the same time, the Navy Area Theater Ballistic Missile Defense (TBMD) Program recognized a similar need. The Navy TBMD strategy involves a layered defense. Navy Area provides close-in coverage, mainly of the battle group and immediate vicinity, with intercepts occurring as threats descend through the atmosphere. The NTW Program extends the protected area to an entire theater of operations and intercepts threats before they have a chance to reenter the atmosphere. These separate programs share similar goals, and many of the system elements, particularly on the ship, are common. As a result, APL decided to coordinate the Area and NTW (or Theater) efforts in the APL Area/Theater Engagement Missile Ship Simulation (ARTEMIS). (Artemis was the Greek goddess of the hunt and protector of children; her arrows never missed.) This article focuses on ARTEMIS-T, the Theater version for NTW.

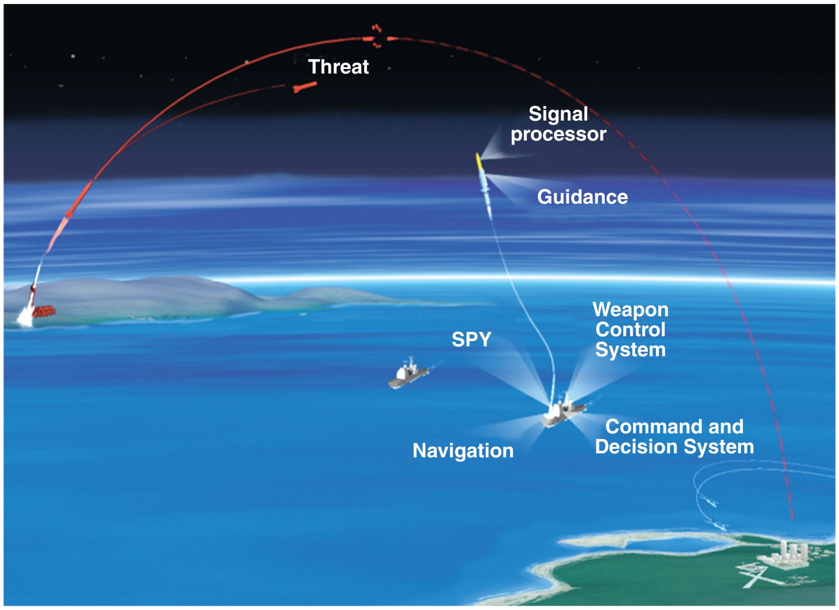

**Figure 1.** NTW problem domain and major system elements.

## ARTEMIS DEVELOPMENT

### Identifying a Strategy

In laying out a simulation development strategy, the first goal was to leverage the substantial investment in existing high-fidelity simulations. APL's core components—

FirmTrack, SM 6-DOF, and BLAST—were obvious from their role in previous analyses.

FirmTrack is a model of the AN/SPY-1 radar carried by all Aegis ships. It schedules and sends dynamically selected radio-frequency (RF) waveforms and then processes the simulated returns to form tracks that are associated with threat launch events and ultimately target discriminate. The SM 6-DOF simulations represent missile guidance with a high-fidelity physics-based characterization of the aerodynamics, control surfaces, rocket motors, and inertial guidance and navigation. They serve as a family of simulations covering the SM evolution and are the certified government standard. Finally, BLAST provides a high-fidelity model of missile sensing and processing. It receives simulated infrared (IR) radiant intensity measurements, processes them into tracks, associates the IR tracks with radar tracks transmitted from the ship, and determines a target object and desired aimpoint. These simulations were the starting point for developing the end-to-end capability in ARTEMIS.

Each of these simulations is a powerful stand-alone analysis tool. The development strategy recognized the need to maintain stand-alone capabilities while adding integrated end-to-end analysis. Further, it was important that for configuration control purposes a single set of code should support both of these goals. This suggested distributed simulation, allowing physically distributed code to participate in a common execution. Distributed simulation also offered the opportunity to expand ARTEMIS beyond its initial capability, allowing for potential future versions to include the prime system contractors' or other government and laboratory simulations.

Given a distributed strategy with potential participation by outside organizations, a high-level architecture (HLA)[1] was a clear choice for implementation. Developed under the auspices of the Defense Modeling and Simulation Office (DMSO), HLA's stated purpose is "to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by the DoD." The Under Secretary of Defense for Acquisition and Technology (USD(A&T)) mandated the use of HLA as the standard technical architecture for all DoD simulations in 1996. In September 2000, HLA was approved as an open standard through the Institute of Electrical and Electronic Engineers (IEEE) Standard 1516.[2] ARTEMIS has followed the HLA Federation Development and Execution Process (FEDEP) and implemented distributed protocols using the HLA RunTime Infrastructure (RTI).

## Executing the FEDEP

The FEDEP describes a generalized model for developing HLA federations. A federation is an HLA term signifying a unified simulation environment bringing together distributed simulation systems to form a new application. Each separate simulation functional component is termed a federate. The FEDEP is a six-stage process of

1. Defining the federation objectives
2. Developing a conceptual model
3. Designing the federation
4. Developing the federation
5. Integrating and testing
6. Executing and producing results

The ARTEMIS-T federation has followed this outline, although not always with a great deal of formalism. Because the ARTEMIS-T team works closely within a single organization, it was possible to reach a common understanding with a relatively brief set of objectives and a simple conceptual model. The primary objective is to provide a high-fidelity integrated end-to-end simulation of the NTW system as a systems engineering tool for performance assessment, risk reduction, concept design, etc. The conceptual model, a representation of the real-world domain as applied to federation space, focuses on the broader objectives. ARTEMIS-T focused on initial NTW capabilities and limited that focus to engaging a relatively simple threat with a single intercepting missile. The conceptual model will be expanded as the system continues to evolve.

### *Selecting ARTEMIS Federates and Allocating Functionality*

Armed with objectives and a conceptual model, the ARTEMIS-T design process began with the selection of federates and the allocation of functionalities among them. Federate selection is driven both by the federation conceptual model of the problem domain and by the more pragmatic question of existing simulation capabilities. The Theater problem domain consists principally of target complexes, ships, and missiles. Relevant ship functionality falls into four primary areas that define the ship-based federates: the navigation system, the AN/SPY-1 radar system, the Command and Decision (C&D) System, and the WCS. Missile functions can be broken down into a guidance and a signal processor federate. A threat federate and a scenario manager for controlling and recording execution complete the ARTEMIS-T federates.

The existing APL simulations described earlier form the basis for many of these federates, although both new and upgraded capabilities are required to reach the initial NTW capability goal defined in the conceptual model. Figure 2 shows the ARTEMIS-T federates and illustrates the notional information flow. All data exchange occurs via the RTI, but for illustration purposes the information flow is shown directly between federates. A brief description of these federates and their responsibilities follows.

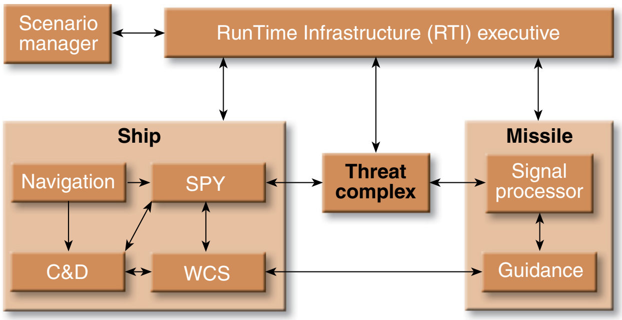

**Figure 2.** ARTEMIS federates and information exchange.

The NTW system is designed to counteract a set of ballistic missile threats. In a sense, the threat is the primary driver of the system. In ARTEMIS-T, the threat federate is based on the object-oriented threat environment developed to populate the BLAST simulation. It models one or more threat complexes, including major objects (e.g., boosters and reentry vehicles) and the associated debris from separation events and fuel venting. This federate provides both position data and selected signature information for every object it models. Providing a common threat picture is a key advantage of an integrated end-to-end simulation.

The ship systems begin with the navigation federate, which is responsible for providing both true and estimated position, velocity, and attitude of the ship. The baseline for this federate is derived from the Area variant of the SM 6-DOF simulation, which requires a ship model to properly initialize the missile. Ship navigation is made a federate to provide a common understanding of ship positional data to the missile, WCS, C&D, and radar federates.

The radar federate is responsible for searching out, tracking, and identifying potential threats. It is based on the FirmTrack simulation, which models the performance of the AN/SPY-1 series of radars with high-fidelity representations of surveillance, detection, radar cross-section estimation, tracking, and discrimination (the process of determining which tracks represent targets of interest). TBMD capabilities implemented in FirmTrack include Linebacker, Navy Area, and ALI designs, with upgrades in progress representing the current NTW design. Once the radar has discriminated a potential TBMD target track, it notifies C&D.

In the ARTEMIS context, C&D is responsible for initiation and management of TBM threat engagements. This represents a small subset of the overall Aegis command and control systems and functions. The required C&D functionality represents a new capability developed for ARTEMIS. The focus for this federate is twofold: performing interceptability checks on potential targets and issuing engagement orders to the WCS.

WCS responsibilities include target track filtering, target engageability testing, missile launch scheduling, ship-to-missile uplink data formation, and missile midcourse guidance commands. The WCS federate is derived from the ALI version of the SM 6-DOF simulation. Since the WCS guides the missile during early stages of flight, the SM 6-DOF simulation contains the required functionality to command the missile.

Missile guidance is responsible for flying the missile to the target, including initialization, boost/pitchover, midcourse guidance, handover, and terminal guidance. The missile guidance federate is derived from the ALI SM 6-DOF simulation and upgraded to represent the NTW system. It receives target data initially from the WCS, then, in the terminal stages, from the onboard missile signal processor.

The missile signal processor's primary responsibility is target and aimpoint selection during the final stages of flight. Based on the BLAST simulation, this federate generates measurements from an IR sensor, forms them into tracks, associates the tracks with radar tracks from the ship, discriminates a target object, and selects an aimpoint on that target. This information is sent to the guidance federate which is responsible for directing the missile to the selected aimpoint.

Coordinating all of the federates is a scenario manager, which is responsible for providing initialization information concerning the situation being modeled. It also tracks the data exchange during execution and can provide a simple visualization. Ultimately the data can be collected and logged by this federate for use in analysis or replay.

### *Defining the Federation Object Model*

The data being exchanged and the participating federates are captured in the Federation Object Model (FOM). Both the Area and Theater versions of ARTEMIS share a common FOM definition. The ARTEMIS FOM defines the federates and their associated data elements and lays out a set of interactions for exchanging data. The HLA allows several types of data exchange; in ARTEMIS, however, all data exchange is accomplished via interactions. This method seems to best reflect the real-world interfaces among systems. To a large degree, ARTEMIS interactions reflect actual system interface messages, although some, such as truth data, are not available in deployed systems.

Table 1 shows the ARTEMIS federates and the interactions they send and receive. Defining the interactions includes specifying each data element and its type. The FOM is useful to arrive at a common early understanding of system interfaces, but it should not be written in stone. It may need to be modified as development proceeds to reflect new understanding or evolutionary changes to the NTW system.

**Table 1. ARTEMIS federates and their interactions.**

| Federate | Interactions sent | Interactions received |
|---|---|---|
| Scenario manager | Initialization | All other interactions |
| Threat | RF_Characteristics<br>IR_Characteristics | Initialization<br>RF_Dwell<br>IR_Look |
| Navigation (ship) | Ownship_Truth<br>Ownship_Update<br>Ownship_Attitude | Initialization |
| SPY (ship) | RF_Dwell<br>TBM_acquisition_Abort_Complete<br>TBM_acquisition_Accept_Reject<br>TBM_track_Data_B6P3<br>TBM_track_Data_L<br>TBM_track_Report_B6P3<br>TBM_track_Report_L<br>Track_Maintenance_Conflict | Initialization<br>RF_Characteristics<br>Ownship_Truth<br>Ownship_Update<br>Ownship_Attitude<br>Missile_Acquisition<br>Missile_RF<br>Cease_Reporting_Drop_Track<br>TBM_acquisition_Request<br>TBM_acquisition_Termination_Request<br>TBM_track_Designation_Request<br>Track_Data_Request<br>Track_Identification |
| C&D (ship) | Engagement_Order<br>Cease_Reporting_Drop_Track<br>TBM_acquisition_Request<br>TBM_acquisition_Termination_Request<br>Track_Identification | Initialization<br>Ownship_Update<br>Ownship_Attitude<br>Engagement_Order_Response<br>Engagement_Status<br>TBM_acquisition_Abort_Complete<br>TBM_acquisition_Accept_Reject<br>TBM_track_Report_B6P3<br>TBM_track_Report_L<br>Track_Maintenance_Conflict |
| WCS (ship) | Engagement_Order_Response<br>Engagement_Status<br>Handover<br>Launch_Order<br>Missile_Acquisition<br>TBM_track_Designation_Request<br>Track_Data_Request<br>Uplink_Command | Initialization<br>Ownship_Update<br>Ownship_Attitude<br>Engagement_Order<br>TBM_track_Data_B6P3<br>TBM_track_Data_L |
| Guidance (missile) | Missile_Data<br>Transition_Mode<br>IR_Characteristics<br>Missile_RF | Initialization<br>Launch_Order<br>Ownship_Truth<br>RF_Dwell<br>IR_Look<br>Handover<br>Target_Data<br>Uplink_Command<br>Transition_Mode |
| Signal processor (missile) | IR_Look<br>Transition_Mode<br>Target_Data | Initialization<br>IR_Characteristics<br>Handover<br>Missile_Data<br>Transition_Mode |

### *Constructing a Framework*

After developing the ARTEMIS FOM, work focused on developing a common HLA framework for connecting the federates and their underlying simulations to the RTI. The RTI provides a set of services that are used to coordinate federate operations and data exchange during execution. Two different development options were considered: having the federates directly invoke the required calls to RTI services or using a "middleware" program designed to simplify the connection process. After evaluating both options, the former was selected. Because the RTI is still relatively new and continues its rapid evolution, many middleware programs have difficulty staying current. Such programs often have a bit of a research rather than a retail flavor. Although this can have advantages, it also tends to mean fewer user manuals and fewer operational guarantees. The RTI itself is well documented, and the helpdesk is quite responsive, so the ARTEMIS team decided that an extra layer would simply cloud the waters.

The RTI provides six management areas to handle federation execution: federation, declaration, ownership, time, object, and data distribution. The framework implements the required capabilities in these management areas for each of the ARTEMIS federates. In particular, the framework handles the federation management activities of creating and joining, synchronizing, and resigning and destroying the federation. In declaration management, the framework has been customized for each federate to specify the interactions that it will be sending and receiving. Object management includes registering objects and sending interactions. RTI time management plays a critical role in federation execution. All of the ARTEMIS federates are both time-regulating and -constrained, i.e., everyone has to wait for everyone else so that the federates stay synchronized. The RTI allows for both time- and event-driven simulations, and the framework handles requesting time advances. Ownership management and data distribution management are not currently required in ARTEMIS. All of the required RTI management areas are built into the ARTEMIS federation framework.

The effect of the framework is to minimize the RTI learning curve for the federate developers. The hooks into the underlying simulations are isolated into three basic function types: `update`, `process_x_interaction`, and `send_y_interaction`. The update function is where the underlying simulations perform the bulk of their processing, while the process and send interaction functions are where the data are exchanged with the rest of the federation. Again, the goal is to allow the underlying simulations for each of the federates to function stand-alone or as a part of an ARTEMIS federation. By isolating the hooks into the simulations, it was easier to wrap the existing simulations without impacting stand-alone capabilities.

### *Framework Testing*

The federate frameworks also created a convenient method for testing the federation before all of the underlying functionality was present. The federate frameworks included basic flow of control so, for example, the threat federate would respond to an `IR_Look` interaction with an `IR_Characteristics` interaction, but neither of the interactions would necessarily contain "correct" data. The individual frameworks for the federates were then executed over the unclassified internal APL network to verify proper connectivity and interaction exchange. Testing on the unclassified network has included up to eight computers across four different subnets running as many as 10 different federates. Some testing has also been successfully performed using the classified APL network where the fully functional ARTEMIS federation will reside.

Framework testing allowed the ARTEMIS team to identify and correct several issues before full functionality was present. Some network and computer configuration issues were resolved. An RTI timing issue was identified involving "zero lookahead" interactions (zero lookahead refers to the ability of a federate to respond to an interaction without advancing federation time). This type of interaction series occurs, for example, when the radar federate sends an `RF_Dwell` interaction and the threat federate responds with `RF_Characteristics`. Such an interaction is modeled as occurring instantaneously. When using a double-based federation time (as opposed to integer time), occasionally an interaction like this was being sent and never received. The cause was traced to a small time delta automatically added by the RTI at each step. The result was that sometimes the RTI would decide that the interactions were old and would discard them. There were two possible solutions. The first was to go to integer timing with a scale factor to translate federation time into scenario time. This would truncate the small delta added by the RTI and allow zero lookahead to function properly. The second solution was to model the "speed of light" time delay by inserting a small time delta and not using zero lookahead. We elected to use the small lookahead value. Identifying issues early with framework testing allowed solutions to be implemented so as not to slow ARTEMIS development.

Another issue that could be tackled using the ARTEMIS framework was Monte Carlo operation. High-fidelity modeling must account for the inherent uncertainties in the TBMD problem domain; in fact, most high-fidelity modeling needs to represent some random elements. This is typically achieved by running in a Monte Carlo mode where multiple iterations of a scenario are executed, allowing certain elements to vary randomly, and then analyzing the results statistically. The simulations underlying ARTEMIS are all Monte Carlo–based. Executing ARTEMIS in a Monte Carlo fashion requires resetting the times on the federates to zero to begin a new iteration. The RTI requires federate time to increase monotonically. Resetting time requires the individual federates to resign from the federation and rejoin in order to reset their times to zero. Again, the Monte Carlo solutions were implemented before connecting the ARTEMIS framework to the underlying simulations.

### *Integrating Functionality*

While the framework handles the RTI interactions, ARTEMIS functionality resides primarily in the underlying simulations. Establishing the proper connections is critical to achieving federation goals. The difficulty of connecting to legacy simulations can vary significantly, depending on the program structure. The more modular the underlying simulation, the easier it is to push data into the simulation and pull the required data out. Connections to the underlying FirmTrack, SM 6-DOF, and BLAST simulations required some restructuring of the legacy simulations. Establishing these connections into an HLA framework allowed years of development effort to be integrated and resulted in higher-fidelity results with relatively small integration costs.

In addition to establishing hooks into the HLA framework, the functionality of the existing simulations was simultaneously being upgraded to represent the initial NTW capability. The high-fidelity models represented the ALI system. The initial NTW capability builds on ALI but provides substantially more functionality against more challenging threats. The primary upgrading difficulty is that the initial design of the NTW system is ongoing. In fact, one of the goals of ARTEMIS-T is to evaluate the system performance of the proposed design to verify that it satisfies requirements. This necessitates close contact with the prime system contractors to confirm preliminary design details as soon as possible. At this point, APL has many of the basic designs and is in the process of upgrading the simulations; however, many of the details on parameter settings, discrimination training sets, and threshold values have not yet been determined. These will be incorporated into ARTEMIS-T as they are defined.

## ARTEMIS VISION

The ARTEMIS-T preliminary capability represents a first cut at the initial NTW capability and will evolve as the NTW system evolves. ARTEMIS can help define system parameters by investigating their effects on overall system performance. It provides a powerful high-fidelity end-to-end tool for evaluating system tradeoffs, testing advanced algorithms, and assessing design performance. It can also serve as a flight test predictor and postflight analysis tool. As threats evolve and more sophisticated countermeasures are hypothesized, ARTEMIS-T can evaluate how the NTW system will perform. By bringing all of the components together into a single integrated simulation, changes to individual components can be propagated throughout the entire system to evaluate effects not just at the component level but also at the overall system level.

Initial participation in ARTEMIS was limited to internal APL federates. As NTW Mission TDA, APL needed an independent evaluation capability. Also, since subject matter experts existed across all system components, design, development, and testing were streamlined within a single organization. APL recognizes, however, that outside organization participation in the ARTEMIS federation could be a valuable form of information exchange within the NTW and Navy Area programs, enabling component developers to evaluate their designs within a high-fidelity system context. Discussions with the contractors about the ARTEMIS federation and potential collaborations are ongoing and promising. The Area TBMD Program has initiated a related effort to build an HLA federation with the Lockheed Martin Medusa radar model using some of the lessons learned from ARTEMIS development. Distributed simulation allows the federation to be opened for participation by the larger development community.

In addition to inviting the greater TBMD community to participate, ARTEMIS will be undergoing several planned upgrades. The initial conceptual model was intentionally limited in scope to allow the federation to be implemented without being overwhelmed by scale. For example, as noted earlier, initial implementation focused on a single incoming threat complex engaged by a single outgoing missile. ARTEMIS will be expanded to encompass raid scenarios where multiple threat launches occur and must be engaged by multiple missiles. In addition to discriminating the lethal objects from other elements of the threat complex, raid scenarios require missile target selection logic to determine which of the observed lethal objects are the missile's primary target.

With the expansion to raid scenarios, capabilities will be phased in. The first threat raids will be engaged by a single ship. Eventually plans call for expanding ARTEMIS to include multiple ships. While simply adding more sets of ship federates (or sets of ship objects) is straightforward, the algorithms for coordinated firing efforts among the ships are more complicated. The Laboratory has an existing ship coordination simulation called the APL Coordinated Engagement Simulation (ACES), and ARTEMIS might draw on this capability to add multiple ship functionality. Thus, upgrading to a full raid scenario capability is a preplanned ARTEMIS improvement.

The capability to visualize ARTEMIS scenarios is also a priority. Previous experience has shown that visualization can be a powerful systems engineering tool in its own right. Outputs from each of ARTEMIS's underlying simulations have been fed to a high-fidelity 3-D visualization program in the past. Similar postexecution visualization can be performed with ARTEMIS. However, other types of visualization could also prove useful. For example, "instrument panel" type readouts during ARTEMIS execution that would visualize the status of the currently executing iteration superimposed on cumulative Monte Carlo results could be a useful tool. Postprocessing visualization could be enhanced to allow more analytical views of system element details of either a single or multiple iterations. Visualization will be used to facilitate the system analysis performed by ARTEMIS.

Looking further into the future, we would like to use ARTEMIS to perform system analysis extending beyond the TBMD domain. In fact, the early coordination between the Theater and Area programs has already provided a head start on extended capabilities because ARTEMIS has the built-in capability to perform both Threat and Area functions. This could be expanded in two different directions. First, additional ship functions (e.g., ship self-defense) could be represented, providing a better system model for analysis of resource utilization issues. Second, ARTEMIS could be expanded to provide a more complete representation of the TBMD domain by including federates to represent other elements of the Ballistic Missile Defense Organization's "family of systems" such as Theater High-Altitude Area Defense.

## CONCLUSION

ARTEMIS has been able to use HLA to build an integrated, high-fidelity, end-to-end simulation leveraging previous investments in legacy simulations. The initial ARTEMIS capability can be increased to include additional participants from the community, allowing others to benefit from this systems perspective. It provides a growth path to include the effects of additional systems and to model the coordination among them. By starting with a relatively modest conceptual model, ARTEMIS has achieved a useful goal and has provided a springboard for expanded capability.

## REFERENCES

[1] Kuhl, F., Dahmann, J., and Weatherly, R., *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice Hall, Upper Saddle River, NJ (1999).

[2] Lutz, R. R., "Migrating the HLA Object Model Template to an IEEE Standard," *Johns Hopkins APL Tech. Dig.* **21**(3), 337–347 (2000).

ACKNOWLEDGMENTS: The authors would like to acknowledge the contributions of the entire ARTEMIS team, both in the Theater and Area efforts. Their support of ARTEMIS and the underlying simulations have been critical to achieve this end-to-end systems engineering capability. Members of the ARTEMIS team include James Belle, David Bort, Kevin Cropper, Allen Driver, Timothy Endres, David Erickson, Robert Ferguson, James Garcia, William Innanen, Alan Joice, David Kenney, Sandra Kource, Kimberly Lichtenberg, and Robert Patterson.

## THE AUTHORS

ANN F. POLLACK is the Supervisor of the Systems Simulation Section in ADSD's Theater Missile Systems Engineering Group. She is the co-lead for the ARTEMIS federation development and an original developer of BLAST. Ms. Pollack holds B.E.S. and M.S. degrees in electrical engineering and an M.S.E. degree in computer science from The Johns Hopkins University. Her e-mail address is ann.pollack@jhuapl.edu.

ANDREAS K. CHRYSOSTOMOU is the Group Supervisor of the Theater Missile Systems Engineering Group and a member of the APL Principal Professional Staff. Mr. Chrysostomou is the co-lead for the ARTEMIS federation and heads a number of TBMD discrimination-related efforts. His background includes extensive experience in command and control systems, signal processing, target tracking, and multisensor data fusion. He has an M.A. in applied mathematics from the University of Maryland. His e-mail address is andreas.chrysostomou@jhuapl.edu.