03F-SIW-100

A Standard Simulation Framework to Support Operational Evaluation of Ship Self Defense

Richard Reading
VisiTech, Ltd.
535A East Braddock Rd.
Alexandria, VA 22314
703-622-8529
reading@visitech.com

Ronald Sawyer
NAVSEA 061R
Naval Sea Systems Command
1333 Isaac Hull Avenue SE
Washington Navy Yard, DC 20376-2305
202-781-2214
sawyerr@navsea.navy.mil

Keywords:
Test and Evaluation, Navy Ship Self Defense, Combat Systems

ABSTRACT: *The Navy Probability of Raid Annihilation (P_{RA}) Assessment Process is a simulation-centric process for operational assessment of ship self defense combat system performance. For various reasons, live testing of end-to-end integrated hardkill/softkill performance against anti-ship cruise missiles continues to be problematic. The Navy P_{RA} Assessment Process leverages federated simulations of ship combat system elements against independent threats in a common environment to augment live results and formulate an overall combat system assessment. A standard federation framework has been implemented in the Navy P_{RA} Simulation Testbed. The P_{RA} Testbed architecture defines the standards for interfacing combat system element simulations, implementing common threat and environment representations, and realizing integrated hardkill/softkill scenarios. Build 2 of the P_{RA} Simulation Testbed deployed the federation across a secure WAN among three U.S. sites and was successfully completed in April 2003. This paper describes the Testbed architecture and its impact on Navy P_{RA} Assessment process standards.*

1. Introduction

Beginning in 2000, the U.S. Navy's Ship Self Defense Combat Systems Engineer has led the development of a common, consistent process for ship combat system operational evaluation. The key measure of effectiveness (MOE) for evaluation of ship self defense performance is the Probability of Raid Annihilation (P_{RA}). The P_{RA} MOE is an assessment metric for the combat system as a whole, measuring the collective performance of the various sensor, control, and engagement elements working together as a unit. For various reasons—technical, safety, and cost—the assessment of ship self defense combat system performance, and of P_{RA} in particular, continues to be problematic in live testing venues.

The Navy P_{RA} Assessment Process was established to address these issues, with modeling and simulation in a prominent role. Previous papers have described the testing issues and process origins in greater detail[1,2]. Technical leadership for process development and maintenance resides with the Ship Self Defense Combat Systems Engineer, now under PEO Integrated Warfare Systems. Important support for development of the Process Standards and Architecture (PS&A) and the P_{RA} Assessment Simulation Testbed has been received from the Navy Modeling and Simulation Management Office and the DoD Director of Test and Evaluation.

2. Characterizing Combat System Performance

The P_{RA} MOE is the capstone MOE for ship self defense. However, there are multiple objectives for implementing the process to assess this MOE. These multiple objectives highlight the fact that the P_{RA} score is less important than why that score occurs:

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **A Standard Simulation Framework to Support Operational Evaluation of Ship Self Defense**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **VisiTech Ltd,535A East Braddock Road,Alexandria,VA,22314-5884**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

- Provide P_{RA} ship class results to meet operational test and evaluation requirements across ship classes in a consistent and adequate manner
- Provide combat system performance insight to the Program Offices and the Ship Defense Combat Systems Engineer
- Provide system capabilities and limitations as inputs for Fleet tactics development

First and foremost, the P_{RA} Assessment Process provides the P_{RA} ship class results to meet operational test and evaluation requirements across ship classes in a consistent and adequate manner. However, there is much more to be gained from P_{RA} assessment aside from the actual score. A second objective of the process is to provide system capabilities and limitations as inputs to Fleet tactics development. That is, helping the warfighter understand how to defend the ship. Finally, the assessment process should provide combat system insight to the PEOs, Program Offices, and the Ship Self Defense Combat Systems Engineer. This insight into integrated combat system performance drivers is perhaps the most important objective, as it enables design and development of future combat systems to meet the evolving threat. This insight to integrated combat system performance is required irrespective of the existence of the P_{RA} MOE.

3. Common Simulation Framework for Assessment of Ship Self Defense

The Navy P_{RA} Assessment Process is a sim-centric process, in that the end-to-end ship defense results can only be calculated within the simulation federation. The process implementation therefore requires a defined architecture and set of simulation standards to be successful; hence, the P_{RA} Testbed and PS&A. The simulation framework for P_{RA} Assessment is founded on the following principles:

P_{RA} Assessment simulation execution will utilize interoperable simulations operating on a single runtime infrastructure.

P_{RA} Assessment will not be achieved by a single monolithic simulation, but rather a set of simulations representing the ship, combat system components, threats, etc. The set of simulations will not be executed sequentially or independently. Fidelity requirements for operational assessment are not realized by this level of simulation interoperability. Rather, the set of simulations will execute together during a common runtime across a network. A single execution of an instance of the set of simulations will determine the result of a single ship defense engagement against a single threat raid. Multiple executions of the set of simulations will be employed to determine P_{RA} results. Scenario progression during runtime will be regulated by the runtime infrastructure, to maintain the integrity of the SoS representation. Further, simulations must permit the regulation of runtime execution rate to slower than real-time to accommodate computation-intensive simulations of sufficient detail for operational evaluation. The simulations shall comply with a pre-negotiated interface definition, given by the Federation Object Model and Federation Agreements. The negotiation of the specific FOM and Agreements documents for a specific implementation will be achieved via the simulation systems engineering process (IEEE 1516.3 FEDEP).

Common, consistent threat and natural environment representations will be achieved through unified modeling with distributed runtime execution.

The Federation Development and Execution Process (FEDEP) calls for the creation of a common system-of-systems object model, here termed the Systems Engineering Concept Model (SECM). It is required to identify all aspects of threat and natural environment that affect any of the systems. This is the unified model that defines the physics that must be implemented in the set of simulations. Allocation of these calculations to individual simulation components will be flexible to accommodate legacy implementations and runtime efficiency, provided they do not violate the integrity of the unified SECM. So, various aspects of threat representation may be distributed among the combat system element simulations, as long as integrated threat representation is ultimately achieved during runtime. For example, threat antenna and body orientation, which impact threat radar signature, may be owned within an EW simulation, while other threat RCS data is owned within a radar simulation, as long as the radar simulation recognizes changes to antenna/body orientation calculated in the EW simulation. Similarly, impacts of natural environment conditions may be calculated in parallel during runtime by individual element simulations, provided they are consistent and recognize changes induced by other element simulations where appropriate.

System-to-system communications should be Interface Design Specification (IDS) and Interface Design Document (IDD) compliant.

This means that tactical system-to-system interfaces (e.g., SSDS to CEP, SSDS to SLQ-32) should be as they are on the ship, to the extent that they impact P_{RA} scenarios. The intent is to avoid re-inventing interface definitions that already exist and to build-in confidence in the resulting systems simulation interface. Further, software testing may be improved by leveraging existing diagnostic tools already aligned to the IDS/IDD. The requirement is for the system-to-system communications to comply with the IDS/IDD, however, the entire IDS/IDD does not necessarily have to be populated in the interface. Only those aspects affecting P_{RA} are required.

System-to-system interactions (e.g., signal propagations, radar reflections, emission detections) should be physics-based through the common environment.
All system representations must be implemented for the SoS environment, and reflect influences and impacts of the other systems present in the combat system. System representations must be implemented to address threat raids rather than one-on-one engagements. Interactions must be represented in sufficient detail to justify accreditation for use in operational evaluation. This will incur a necessary runtime execution pace of slower-than real-time, to accommodate computation-intensive physics-based calculations. Therefore, the simulations must permit the regulation of runtime execution rate to slower than real-time.

4. P_{RA} Simulation Testbed Architecture

The fundamental purpose of the P_{RA} Assessment Simulation Testbed is to create a working simulation framework that meets the foundation requirements. It is a tangible product of the Navy P_{RA} Assessment Process development, and it represents a proof-of-concept tool for the Navy P_{RA} Assessment Process approach.

The Testbed is an important asset to support Ship Class Program Manager (PM) execution of P_{RA} assessment, for several reasons. It is being used as a source of standards for use by element Program Managers in developing system models needed for instantiating an integrated combat system representation. The Testbed creates a simulation infrastructure for element PMs to test their models in a system-of-systems setting. It provides common services to eliminate redundant model development and enable consistent re-use of system representations across ship classes. It is an ideal platform for simulation risk reduction starting at the element level where system component models can be tested prior to delivery to the ship class PM. The ship class PM can use the Testbed to retire simulation risks early in P_{RA} assessment process execution. Further, the use of a common simulation infrastructure improves validation confidence and efficiency. Thus, the P_{RA} Simulation Testbed reduces risk and increases SoS representation fidelity.

The baseline Testbed, Build 1, was an initial implementation of the interoperable simulation architecture required for P_{RA} assessment. Testbed Build 1 was a rapid prototype development during the latter half of CY 2001. It was integrated on a classified LAN at the Naval Research Laboratory, Washington DC (NRL DC). The simulation components and functional allocation for Build 1 are depicted in Figure 1. Testbed Build 1 mapped to the LPD 17 combat system configuration as a use case. Results from Testbed Build 1 execution were demonstrated in January 2002.

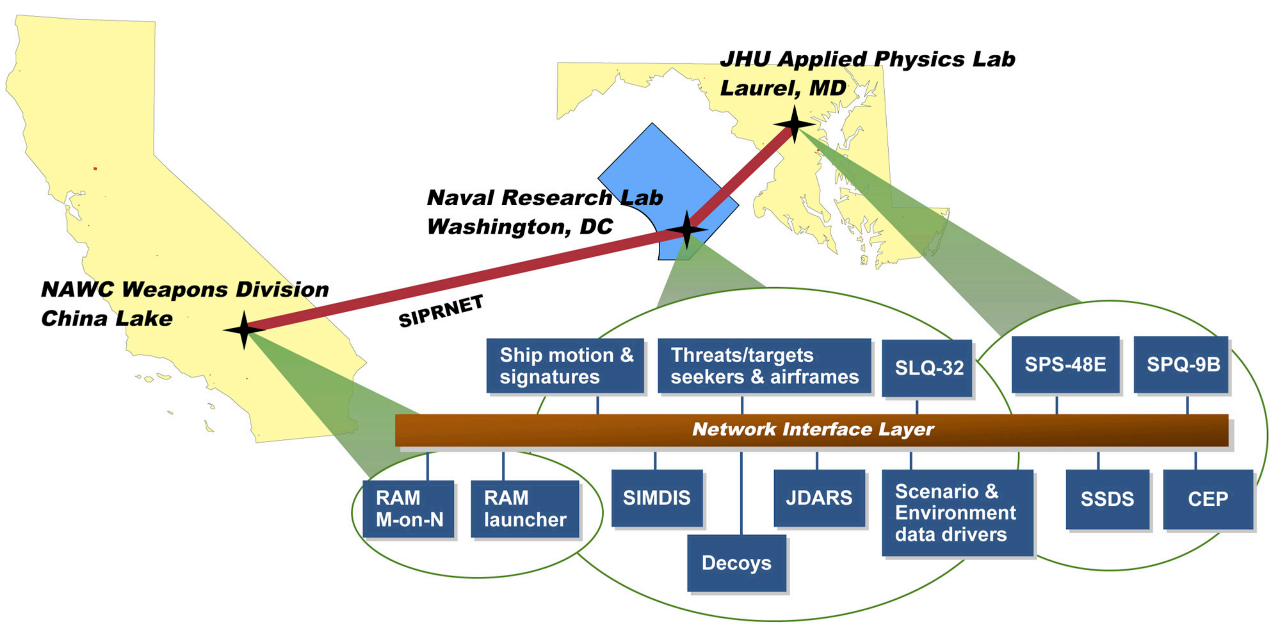

Figure 1. P_{RA} Simulation Testbed Build 2 Deployment.

P_{RA} Assessment Testbed Build 2 evolved the Build 1 capability by distributing execution across a classified WAN. Network connectivity was achieved using SIPRNET connections among sites at NRL DC, JHU/APL, and NAWC Weapons Division China Lake, CA (see Figure 1). Testbed Build 2 also improved representations of the RAM missile and included more sophisticated scenario data distribution. Results from Testbed Build 2 execution were demonstrated in April 2003.

The P_{RA} Simulation Testbed Architecture has now transitioned to its first operational implementation by the LPD 17 amphibious ship class for use in their ship class operational evaluation. Future ship classes are anticipated to follow suit.

4.1 Re-hosted Tactical Software

Runtime execution details for the Testbed architecture have been previously described. However, implementation of the CEC element is worth additional discussion. The CEC representation for the Testbed was achieved using re-hosted tactical code. The tactical code in the operational CEC runs as real-time embedded software. For the P_{RA} Testbed, the code was re-hosted to a general purpose workstation and interfaced to software layers to handle process calls and clock progression. By using these 'adaptation' and 'time management' layers, the team was able to fool the tactical code into thinking it was running real-time. When implemented in the Testbed, the CEC tactical code ran as a time-regulating and time-constrained federate, while still executing the operationally correct sensor fusion algorithms and tactical communications with the SSDS and radar federates.

Transferring tactical communications data across the RTI is actually handled in a rather simple fashion. Each tactical data packet passed from one system to the next is treated as an opaque data interaction for the RTI to move between the two system federates. The contents of the data packet not defined in the FOM, as they are already defined in the appropriate IDS/IDD (see Figure 2). This approach keeps the FOM information sparse and easily managed, and permits the system federate developers to use existing references they are already familiar with to develop their federate interfaces. Of course, the drawback is that the requirement for deciphering the data contained within these interactions resides with the recipient, making it more difficult for third-party federates to make use of them. This can be troublesome in situations where a small portion of a data packet may be useful to multiple federates, possibly causing that portion to be duplicated in a separate data publication for general federation use.

Portions of the SSDS Mk 2 representation for the P_{RA} Testbed were also developed using re-hosted tactical code (SSDS Mk 1 software was successfully re-hosted during a previous HLA demonstration project[3]). The experience with the CEC and SSDS federates has been so positive that future implementations of SLQ-32 and radars may follow a similar approach, for example re-hosting SPQ-9B radar tracker code in lieu of a model. This approach of re-hosting tactical code is enticing for operational evaluation, because it is fairly non-intrusive and eliminates the need to 'model' algorithms contained in the code.

Interaction	Parameter	Data Type	Units
CEPBuffertoSPQ-9B	data	any	N/A
CEP_TCPBufferToSPQ9B	data	any	N/A
CEP_TCPBufferToSPS48E	data	any	N/A
CEPBuffertoSPS-48E	data	any	N/A

See WS 33551 Interface Specification and SPQ9B-ids.h for message contents.

See WS 32905 Interface Specification and SPS48E_ids.h for message contents.

Figure 2. P_{RA} Testbed Tactical Systems Communications FOM example.

5. P_{RA} Testbed Products and Lessons Learned Thus Far

Development of the P_{RA} Testbed Builds 1 and 2 have produced significant results, both in the form of tangible products and lessons learned. The culmination of these products and lessons is significant risk reduction for future implementations of the Navy P_{RA} Assessment Process, and therefore less risky and costly ship class operational evaluations.

5.1 Products

The Navy P_{RA} Testbed development team has undergone the experience of implementing two spirals of the development process for a ship defense simulation federation. In doing so, they have generated the various FEDEP system engineering products for re-use, most significantly:

- Ship defense Systems Engineering Concept Model
- Ship defense federation functional allocation
- Federation Object Model
- Federation agreements

The team has tested the common simulation framework requirements and exercised the interoperable simulation architecture, both with successful outcomes. They have developed the initial set of simulation modules and support tools, and established secure wide-area network connectivity for executing P_{RA} analyses.

Perhaps most importantly, the P_{RA} Testbed Build 2 implements for the first time integrated hardkill and softkill element representation in the same runtime infrastructure with a common, reactive threat raid representation.

5.2 Selected Lessons Learned Thus Far

As is normally the case with prototype implementations, there are a plethora of lessons learned garnered from P_{RA} Testbed development. A selection of interesting and important lessons follows:

FEDEP importance.
Testbed development has highlighted the important role of early stage systems engineering as called out in the IEEE 1516.3 Federation Development and Execution Process. This includes heavy emphasis on cross-element negotiation and conceptual modeling, and is particularly important for achieving consistent, credible threat and natural environment representations.

Re-hosting of real-time tactical code.
As previously discussed, the CEP and SSDS representations implemented in the P_{RA} Assessment Testbed utilize tactical code re-hosted in a workstation environment. Common Adaptation Layer and Time Server software was used in the CEP and SSDS federates to handle calls and control time perception for the tactical code. All interfaces between the CEP, SSDS, and the other shipboard elements (radars, SLQ-32, RAM) comply with the appropriate IDS/IDD definitions.

Due to the use of a common Adaptation Layer, the re-hosting of tactical code was made possible in the short time available between inception and the Testbed demo. A similar re-host of SSDS Mk 1 tactical code for the PEO TSC HLA Pilot Program, before the Adaptation Layer was available, was much more labor intensive and time consuming.

Slowing processing down from real-time.
Much of the development time was used to develop and refine a Time Server software clock package to control time for the re-hosted tactical code. All processes/tasks/threads within the re-hosted real-time code functions of a particular federate had to be synchronized with each other in order for the re-hosted real-time code to run under RTI's time-regulated/time-constrained synchronization paradigm. It was discovered that task delay requests less than the RTI's default time request and grant cycle time cause inefficiencies which slow federation execution. Additionally, scheduling of multithreaded applications may change from their native hardware environments. Best multithreaded software development practices should not rely on a particular scheduling of threads for proper execution. However, especially for legacy code generated without emphasis on multithreading techniques, a simulated hardware delay can be implemented.

Implementing legacy models in an HLA/RTI framework.
The use of Interface Design Specifications (IDSs), where applicable, as a guide to Federation Object Model (FOM) development shortened the FOM development time. Deviations from IDS content and format dramatically increased FOM development time in the earlier HLA Pilot federation.

Reactive threat representations are viable for integrated HK/SK scenarios.
This is essential for P_{RA} assessment, and was achieved in both Testbed Builds 1 and 2, wherein outgoing RAM missiles homed on threat ASCMs that were being influenced by Nulka seduction tactics. Both radar federates also subscribed commonly to this reactive threat information, so they could respond dynamically to, for example, changes in threat spacing that could affect the ship's ability to establish a correct raid count.

Experience is essential.
Prior experience was critical in working the timeline that was established for each Testbed build. By leveraging previous experience in developing interoperable simulations, a working prototype of an integrated combat system representation was achieved in under six months in Testbed Build 1. Work that had gone before, particularly in HLA development and in embedded system re-hosting, reduced risk and made the Build 1 effort feasible under such formidable time constraints. Build 2 implemented an even tighter federate development and federation testing cycle, leveraging a consistent set of developers from Testbed Build 1. Maintaining a stable core Navy team will be important in

the future for ensuring consistent and efficient Testbed implementations across ship classes.

LAN to WAN transition.

The transition from LAN to WAN was relatively easy, partly due to the fact that network bandwidth was not an issue. Since the Testbed normally executed slower than real-time, network performance did not hinder integration or affect execution results.

Optimizing Execution Time.

Familiarity with HLA and the subtleties of its software incarnation, the RTI, is helpful in assuring that functional allocation is optimal and inter-federate communications are as efficient as possible. The Testbed development thus far has been conducted using DMSO RTIs. It will be interesting to see how this situation improves or degrades with transition to commercial RTIs.

6. Summary

The Navy P_{RA} Assessment Process will allow combat system end-to-end assessment not otherwise possible via live test events. The P_{RA} Simulation Testbed is providing the products and lessons learned for evolving the Navy P_{RA} Assessment Process Standards and Architecture. The P_{RA} Testbed is a common framework for integrated combat system representation that enables first-ever integrated hardkill/softkill results against reactive threat representations. The LPD 17 ship class has transitioned the Testbed architecture to support its ship class operational testing. Other ship classes will follow, while element programs can use the common framework to explore system performance in the presence of the complete system-of-systems. The way ahead will see the Navy P_{RA} Assessment Process refine the architecture and modeling standards through Testbed experimentation and development & learning from LPD 17 P_{RA} assessment.

7. References

[1] Reading, R. & Sawyer, R., *Simulation Based Testing of Ship Self Defense* paper 02S-SIW-112, March 2002.

[2] Reading, R. & Pobát, M., *Common Threat Representation in Simulation and Testing of Ship Self Defense*, paper 00S-SIW-129, March 2000.

[3] Zouck, et al, *Porting Real Time Tactical Code to Construct an HLA Federate* paper 99S-SIW-159, March 1999.

Author Biographies

RICHARD A. READING is a senior engineer with VisiTech, Ltd. He is the Systems Engineer for the Navy P_{RA} Assessment Process and the P_{RA} Simulation Testbed. He is also providing technical leadership and support to multi-national HLA federation developments applied to systems acquisition, including the NATO Naval Armaments Group 6 sub-group on Virtual Ships and the NATO Maritime Missile Defense forum. He is also the International Project Team Leader for the NATO Interoperability and Re-Use Study, NIREUS. Mr. Reading served as the system engineer for the Navy PEO Theater Surface Combatants HLA Pilot Program. Previously, he spent over ten years at the Naval Research Laboratory developing hardware and software simulations for surface ship electronic warfare systems development.

RONALD J. SAWYER is Warfare Systems Engineer for Architecture, Standards, and Policy with NAVSEA 06. He was previously the Deputy Combat Systems Engineer for Ship Self Defense at the Navy Program Executive Office for Integrated Warfare Systems (PEO IWS). He was the project manager for the Navy P_{RA} Assessment Process and the P_{RA} Simulation Testbed development. He was also the project manager for the Navy's Self Defense Test Ship and Requirements Analysis Working Group. Mr. Sawyer has also served as the Director of Modeling and Simulation for PEO TSC and Systems Engineer for Test Assets Development for the Cooperative Engagement Capability Program Office. Mr. Sawyer has over twelve years of experience in Test and Evaluation, including two and a half years working as an operational test analyst for OPTEVFOR.