Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

SIMULATION ENVIRONMENT TO ASSESS TECHNOLOGY INSERTION IMPACT AND OPTIMIZED MANNING

Niraj Srivastava

The Sevaan Group LLC.
411 North Lane
Prince Frederick, MD 20678, U.S.A.

Gary Horne

Referentia Systems Inc.
550 Paiea St
Suite 236
Honolulu, HI 96819, U.S.A.

Frank Pietryka

The Sevaan Group LLC.
411 North Lane
Prince Frederick, MD 20678, U.S.A

Mark Theroff

NSWC Crane
Code 6048, Building 2035
300 Highway 361
Carne, IN 47522, U.S.A.

ABSTRACT

The reduction in life-cycle costs for Naval vessels is critical for operating a cost efficient and robust Navy. Computer based simulations are an effective tool for human system integration optimization, as well as for studying the risks associated with complex interaction between crew and systems. The proposed modular simulation environment empowers analysts to choose and integrate the best combination of agent, discrete event, and physics based simulations to address questions of manning. The environment embraces advances in complexity theory for simulating non-linear systems, knowledge discovery for data analysis and distributed computing for execution environment.

1 INTRODUCTION

This paper describes an environment for manning and technology optimization that leverages many of the technologies and methods created by Project Albert and the open source community. Project Albert is a program started in 1998 and based at the Marine Corps Warfighting Laboratory (Horne 2001, 2002, 2003). A key goal of our study is to develop a feasibility prototype for the navy. This foundation work is essential to ensuring the resulting feasibility prototype will demonstrate immediate value by providing the capability to both estimate the relationship between a technology and shipboard manning as well as to compare the manning requirements associated with sets of competing technologies. For this reason, this project focuses on identifying component structures that are required for an overall solution. We have constructed a synthetic environment for model development, for run-time and post-run-time analysis of interactive, multi-agent applications. This paper presents the motivation for creating a generic extensible toolkit and describes the framework we have developed with a prototype that works with complex agent system.

2 BACKGROUND

Manning costs are the single largest expense incurred over a ships life cycle (GAO-03-520). Facilitating reductions in total ownership costs are paramount in operating a cost efficient Navy. Crew manning must be optimized for lifecycle costs including compensation, training, health and safety, habitability, recruitment, and retention, but balanced with an acceptable risk of service (Bost and Galdorisi 2004). Computer based simulations are an effective tool for human systems integration. A synthetic environment that can be used to estimate the relationship between a technology and shipboard manning, as well as to compare the manning requirements associated with sets of competing technologies, can be of great value to Naval planners.

Designers today are faced with the challenge that a ship's service life is expected to exceed many decades, yet technology is changing at a double exponential rate (Kurzweil 2001). Designers can only guess at future technical innovations that will result from the constant co-evolution of military tactics and future threats. Simulations must address the risks associated with the complex interactions of personnel and systems of systems as it is the unlikely occurrence of unplanned events that can lead to catastrophe. Simulations must also embody the human behavior based upon cognitive and behavioral psychology. It is the human element that produces surprises in unforeseen situations and has the ability to overcome nearly disastrous events through actions that may not be in the “event graph” of “the actor” in conventional simulations.

The challenge of simulating human interaction with technical systems is that while systems may be unpredictable they are deterministic; however, human behavior is nondeterministic and unpredictable. It is beyond the limits of the current technology and mathematics to develop computable algorithms for nondeterministic and unpredictable systems. However, systems with nonlinear interactions can be modeled and computed to study complex dynamics and unpredictability. Nonlinear systems with interesting emergent behavior are often referred to as complex systems. Additional complexity arises when the components of the system can change and evolve over time. Systems with this additional property are sometimes called complex adaptive systems (CAS). Reasonable models of systems consisting of humans and machines are by nature CAS.

Building models of CAS is difficult due to nonlinearities and evolving behavior of the component elements of the system. Furthermore, detailed simulations are problematic because it is virtually impossible to get all of the details correct. Traditional Discrete Event Simulations (DES) are effective tools for modeling deterministic systems such as weapon systems, radar systems, navigation systems, etc. Humans on the other hand are not always modeled as finite-state machines. It has been said that “Exact, mathematical military calculations have no firm basis in war” (Koenig 1998). The use of autonomous agent-based simulations is an effective method to study CAS and provides a means for simulating emergent human behaviors that lead to the nonlinear phenomenon experienced during the fog of war (Davidsson 2000). Estimating manpower is as much about the intrinsic, such as morality and leadership, as it is an individual’s ability to perform a given task.

3 SIMULATION ENVIRONMENT

A simulation environment for studying complex system behavior is based upon the following architectural components:

- User interface: guide the user through the workflow of design scenarios, sampling parameter space (experiment design), simulation engine, data analysis and visualization, using natural language engine
- Modular design: allow flexibility to adopt rapidly evolving technologies in simulation, data analysis, user interface, knowledge discovery
- Resource management: standard based distributed computing infrastructure to allowing scaling of simulation to 100000s of nodes
- Knowledge Discovery tools: for data analysis and simulation guiding.

The non-linear nature of human interaction precludes a single computer simulation from providing the capability to completely predict the actions of a ship’s crew. Additionally the natural evolution of technology is such that it is always better to provide a framework that allows the use of multiple technologies vs. a dependence on a single implementation. The synthetic environment is designed with this evolution in mind. Instead of relying on a single simulation, the environment is designed in a modular manner such that new simulations may be easily introduced in the future as newer more robust models become available.

The core components of the environment include an event manager, one or more agent based crew simulations and one or more discrete or physics based sub-system simulations.

Event Manager. The event manager is responsible for coordinating simulation execution. By design, the synthetic environment needs to support hybrid agent-based simulations consisting of a mixture of systems (discrete and continuous). A key to representing the interaction of crew and technology is the synchronization in time of the interacting simulations. For example, a radar system simulation may be operating in microsecond fidelity while the crew simulation may operate in intervals of seconds. Additionally a discrete event system may use events to drive time intervals. Instead of forcing different simulations and agents to a particular time stepping scheme which could lead to inefficiencies and convergence issues, the synthetic environment uses an event manager that requires all components of the simulation architecture to use a standard interface. The event manager using the standard interface coordinates all activity and information between components and agents. Another important function of the Event Manager is to manage an interaction between a crew member and a technical subsystem. These I/O flows are implemented through a generic interface.

Crew Simulator. One or more autonomous agent based simulations are used to simulate the actions and interactions of a crew. At the fundamental level the simulation is comprised of independent agents each representing a crew member. The actions of each crew member are governed by a rule based decision process as they interact with the environment incrementally in time. Each agent possesses the fundamental abilities common to all sailors from the ability to traverse the vessel to the ability to represent behaviors such as eating and sleeping. Agents also possess expertise based on their duty assignment. For example, sailors assigned to the weapons department, such as Sonar Technicians, possess expertise in the ASW suite utilized to detect submarines while sailors in the combat system department, such as Fire Control Technicians, are responsible for air and surface sensors integrated with missile systems. The simulation includes parameters of effectiveness skills based on training, experience as well as intrinsics such as morale, stress, and fatigue. Agent based simulations are also useful in simulating the effects of leadership on performance.

Systems and Subsystem Simulators. Technology on naval vessels is commonly described bysystems and their corresponding subsystems and, when integration occurs, as systems of systems. They are grouped in terms of functionality such as the theater air dominance system which includes the radar system, cooperative engagement capability, the weapon control system, the advanced integrated electronic warfare system, as well as the maritime dominance, land attack, command and control, and mission support capability systems. A variety of simulation techniques have been implemented for modeling systems level technology. Discrete event and physics based simulations are most common. The synthetic environment includes a suite of system simulations for representing the physical and functional parameters of technical systems.

Figure 1 depicts the high level integration of the core components of the synthetic environment that provide the ability to study performance based on measures of effectiveness for an individual scenario.

Figure 1: An Environment That Integrates Autonomous Agent Based Simulation for Personnel for Simulating Naval Systems Is Optimization

The suite of analysis tools that allow for the automation of execution over large variations in initial conditions and the visualization of multiple resulting scenarios is known as the data farmer. The manning information harvester provides even greater automation of the data farming process by steering the data farming process through the use of genetic algorithms.

Data Farmer – Data farming involves the investigation of a wide number of variables across a wide range of values multiple times. In essence, the user is attempting to model many combinations and variations within the data space and grow resulting data in an iterative process attempting to answer questions at hand. Multiple runs of the same scenario are important to determine a statistically significant representation when working with agent based models due to their nondeterministic and unpredictable characteristics. The data farming environment includes a suite of tools for scenario management, analysis, and visualization. (Horne and Meyer 2004).

Knowledge Harvester (Pietryka 2005) – The data farmer provides the tools for the analyst to perform comprehensive searches (i.e. what-if comparisons) for a variety of measures of effectiveness. While this functionality is essential for thorough analysis, the process can be time consuming for the analyst. The manning knowledge harvester automates the data farming process by using genetic algorithms to manage the execution and analysis of the resulting large multidimensional global datasets where the search space potentially contains multiple local minima. Unlike other search methods, correlation between the search variables is not generally a problem for genetic algorithm based analysis. The knowledge harvester does not require extensive knowledge of the search space, such as likely solution bounds or functional derivatives.

Figure 2 depicts the high level architecture of the fused simulations controlled by the event manager in the data farming environment. The data farming environment provides the functionality for addressing large variations in scenarios on a variety of computing platforms, as well as the visualization and analysis tools. The data farming process is automated by the manning knowledge harvester.

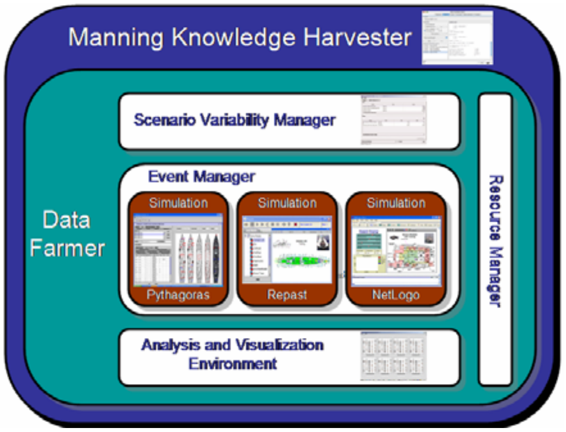


Figure 2: The Manning Knowledge Harvester Automates the Data Farming Process over a Variety of Scenario Parameters Allowing the Analyst to Quickly Address Multiple Measures of Effectiveness.

The core of the synthetic environment is a collection of autonomous agent based simulations. For this study the Pythagoras simulation tool was chosen due to it's ease in prototyping. Pythagoras is an open source agent based simulation written in Java and developed by Northrop Grumman. Pythagoras is a logical choice due to its event-triggered personality capability that allows for straightforward integration with technical system models. The simulation is guided by soft decision rules in order to ensure traceability and to retain some elements of Fuzzy Logic while avoiding the "everything is gray" result. The system is designed to be data farmable and is able to run on a distributed computer for 100,000 or more replicates using XML for input and output.

The other key areas of our simulator are Data Farming and Data Harvesting. We have identified and tested open source components for each areas. For data farming, we have tested Triana. Triana is a graphical problem solving environment, both a problem solving and a programming environment, providing a user portal to enable the composition of simulation. Users compose workflows by dragging programming components, called units or tools, from toolboxes, and dropping them onto the workspace.

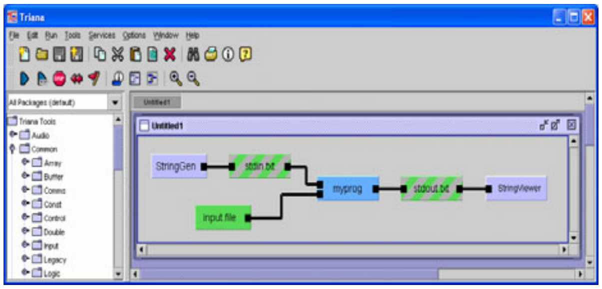


Figure 3: Triana Workflow

The Triana workflow environment is utilized to implement the application for uniform access of local processes, interactive tools and grid-enabled remote services. We have concluded that Triana, developed in Java, provides benefits to flexibility, reusability and scalability and has potential to become a mainstream distributed-computing application enabling technology. And for Data Harvesting, we have tested Weka, a state-of-the-art facility for developing machine learning (ML) techniques and to apply them to real-world data mining problems. Weka has incorporated several standard ML techniques into a software "workbench" called WEKA, for Waikato Environment for Knowledge Analysis. With it, we are able to use ML to derive useful knowledge from results of data farming that are far too large to be analyzed by hand.

Pythagoras can be data farmed in the triana environment using triana's loop elements and the output can be piped to weka or even gnuplot.

4 DISTILATIONS

A number of simple scenarios are used to demonstrate the value of the environment to explore the compelling capabilities provided by new technologies such as wireless networking.

In one example an agent is tasked with traversing a vessel to provide aid to sailors in distress. The resultant data showed a significantly decrease in response time for agents equipped with a wireless geo-spatial locator vs. agents who were lacking this capability. It is expected that these types of simulated results are valuable to the acquisition process.

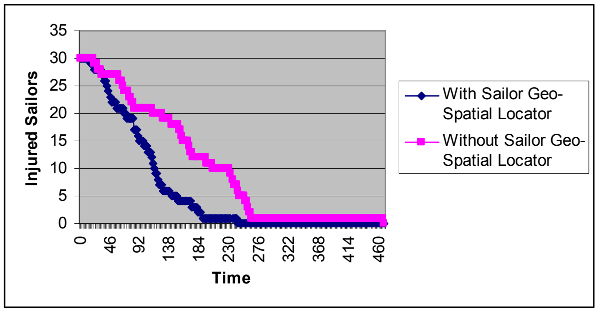
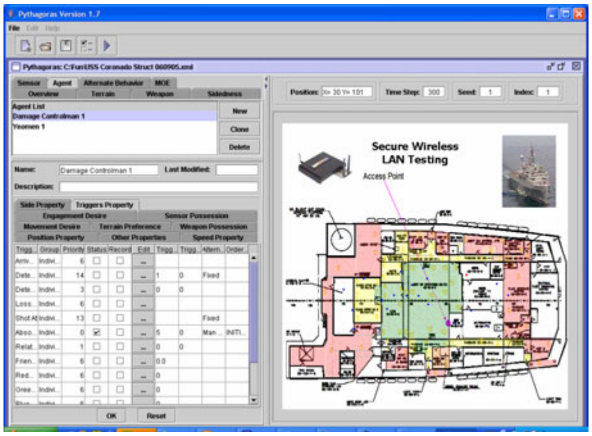


Figure 4: The Result of a Simple Demo to Show the Effectiveness of a Geo-Spatial Sailor Locator

Another example used for testing is intended to demonstrate the ability of crew (agents) to navigate the structure of a AGF class vessel. Crew actions are dictated by a variety of rules and decision making processes. Pythagoras is a time-step driven model. The user indicates the number of time steps to be run for and these time steps determine the agents' behavior. For each time step, the crew member follows a time cycle that includes self-evaluation, sensing the environment, deciding on actions, interacting with crew or technology, moving, and recording.



Agent List
Damage Controlman 1
Yeomen 1

Name: Damage Controlman 1
Last Modified:
Description:

Position: X= 30 Y= 101
Time Step: 300
Seed: 1
Index: 1

Figure 5: Agent Behavior on a DDG Vessel

This scenario demonstrates the use of sailors equipped with geo-spatial locators accessing a wireless network. In this simulation a number of blue sailors have been overcome by something which is causing them to wander aimlessly. The red agent, a hospitable corpsman, is tasked with seeking out each blue sailor and providing care. Once care has been administered each blue sailor proceeds to the rear of the vessel while the corpsman continues to seek out other sailors.

The corpsman has the use of three distinct methods to locate the sailors. The first method is line of sight vision; the second is through a wireless ad hoc connection with a sailor. The third is through a wireless network connection to the geo-spatial locator when a sailor is within range of an access point.

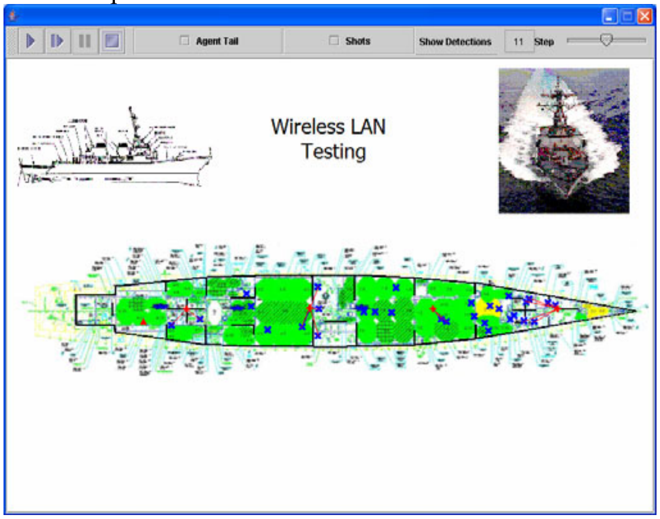


Figure 6: Demonstrate the Capability of Geo-Spatial Locators and Wireless Networks through a Search and Rescue Drill

5 ANALYSIS

While each simulation package has its own analysis capability, the suite of analysis and visualization tools in WEKA allow the analyst to quickly view and compare the results of a large number of simulations simultaneously. The tools presents surface of statistical summary (mean, std dev., quartiles, min, max) from the variation of runs. Tools can display interesting relationships between input parameters and measure of effectiveness (MOE).

6 DISCUSSION

The Defense Acquisition Guidebook states that "the program manager faces a myriad of considerations and management tools to translate the users desired capabilities into a structured system of interrelated design specification. This is clearly not a trivial task. It is an iterative task, performed within the framework of Systems Engineering to achieve the 'best value' for the user. (Defense Acquisition Guidebook)" A variety of tools exist for supporting "what if?" trade studies needed to address the multitude of often conflicting design considerations. Manual and parametric estimation approaches for designing affordable systems are used to estimate cost, effort, and schedule. Manual estimation tools rely on analogy techniques base estimates by comparison of other comparable projects. Engineering buildup tools are used by domain experts who perform engineering "ground up" estimates. Rules-of-thumb tools utilize factors including productivity metrics, percentages, or multipliers applied to size, staffing, or other estimate data. Parametric tools use data collected from numerous actual projects to drive algorithm based estimates.

Traditional approaches are limited in their ability to simulate process efficiency. Operations, maintenance, logistics activities are significantly influenced by the behavior of crew. Behavioral rules that determine the interaction between systems and crew must be captured in any tool designed to investigate process efficiency. Simulations must address the risks associated with the complex interactions of personnel and systems of systems as it is the unlikely occurrence of unplanned events that can lead to catastrophe. Simulations must also embody the human behavior based upon cognitive and behavioral psychology. It is the human element that produces surprises in unforeseen situations and has the ability to overcome nearly disastrous events through actions that may not be in the "event graph" of "the actor" in conventional simulations.

A synthetic environment that can be used to estimate the relationship between a technology and shipboard manning, as well as to compare the manning requirements associated with sets of competing technologies, can be of great value to Naval planners.

REFERENCES

Bost, B and G. Galdorisi. 2004. Transforming coalition naval operations by utilizing human systems integration to reduce warship. *Manning Lessons Learned from the United States Navy DDG-51 Class Warship Reduced Manning Study*. `<http://industries.bnet.com/abstract.aspx?scid=2575&docid=115981>` (accessed December 5, 2004).

Davidsson, P., 2000. *Multi agent simulation: Beyond social simulation*, ed S. Moss and P. Davidson: MABS 2000.

Defense Acquisition Guidebook, `<http://akss.dau.mil/dag/>` (accessed August 11, 2005)

Horne, G. (editor) 2001, 2002, 2003. *Maneuver Warfare Science 2001, 2002, 2003* (series of publications). USMC Project Albert. Quantico, Virginia.

Horne, G. and T. Meyer, 2004. Data farming: Discovering surprise. *Proceedings of the 2004 Winter Simulation Conference*. Denver, Colorado.

Koenig, J. Command and nonlinear science in future war. *Maneuver Warfare Science 1998*, Marine Corps Combat Development Command Publication.

Kurzweil, R. 2001. The Law of Accelerating Returns. <http://www.kurzweilai.net/articles/art0134.html?printable=1> (accessed 15 November, 2004).

Military Personnel. 2003. Navy Actions Needed to Optimize Ship Crew Size and Reduce Total Ownership Costs. GAO Report To Congress 2003, GAO-03-520. <http://www.gao.gov/new.items/d03520.pdf> (accessed 30 November, 2004).

Pietryka, F. and N. Srivastava 2005. Optimizing "What if?". To be published.

Triana, <http://www.trianacode.org/collaborations/> (accessed August, 11, 2005).

WEKA, <www.cs.waikato.ac.nz/ml/weka/> (accessed August, 11, 2005).

AUTHOR BIOGRAPHIES

NIRAJ SRIVASTAVA, Ph.D. is a senior scientist with a degree in physics. His career spans academia to industry. Dr. Srivastava has published in the areas of non-linear dynamic and chaos theory. He has worked as consultant for simulation development, on techniques for scalability and improving performance of massively distributed application. His e-mail address is <Niraj@sevaangroup.com>.

GARY HORNE is the founding member of the Data Farming Team at Referentia Systems Inc. He holds a B.S in Mathematics and an M.Ed. in Education from the University of Maryland and a D.Sc. in Operations Research from the George Washington University. He is the originator of the Data Farming methodology and the Director of Project Albert. His email address is <ghorne@referentia.com>.

FRANK PIETRYKA is a senior solutions architect and software engineer with extensive experience in modeling, simulation and training, image and signal processing, and high performance computing. He has demonstrated success in the development and implementation of corporate software engineering practices and procedures. His e-mail address is <Franko@sevaangroup.com>.