23rd International Conference on Harbor, Maritime and Multimodal Logistic Modeling & Simulation
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0339 ISBN 978-88-85741-59-1 © 2021 The Authors.
doi: 10.46354/i3m.2021.hms.007

Verification of a Naval Logistics Planning Simulator Prototype

John F. Richardson[1,*]

[1]Naval Information Warfare Center, 271 Catalina Blvd, San Diego, 92152, United States

*John F. Richardson. Email: richards@spawar.navy.mil

Abstract

This paper discusses a verification process for a black box Naval Logistics Simulator developed via DARPA research on optimization techniques. The black Box Naval Logistics simulator that is the Simulator System Under Test (SSUT) is the Composer 1.0 Naval Logistics Simulator. The Composer naval logistics optimization environment is a simulation tool for optimization of Naval Logistics related to logistics supply, supply prepositioning and risk-aware planning for logistics. Composer is developed currently for cloud environments using Docker Containers for virtualization in the cloud. Composer has fast algorithms to reduce option space and planning. It is currently approximately ten times faster than traditional optimization approaches. Composer has a very sophisticated graphical user interface for creating logistics scenarios that are optimized during simulation runs. Composer was developed as part of the Defense Advanced Research Projects Agency (DARPA) Complex Adaptive System Composition and Design Environment (CASCADE) research efforts customized for Navy Distributed Logistics.

Keywords: Verification; logistics; Maritime; Simulator

1. Introduction

The Composer 1.0 simulator is a research simulator designed to apply optimization techniques to naval logistics. The Composer 1.0 simulator was created by Metron Inc. It was delivered with optimization libraries and a graphical user interface. The optimization libraries implemented the optimization techniques that would be used in naval logistics simulations. Advanced machine learning optimization techniques were developed and the development of optimization algorithms for naval logistics simulation was the primary requirement. The requirement to develop advanced optimization algorithms resulted in several derived requirements that would be subject to verification and indirectly validation.

One of the simplest methods to provide metrics for optimization results is to provide flat file input and output. Since the goal of Composer 1.0 was a naval logistics simulator that would have to be "useful for the logistics community" and provide a realistic simulation tool for logistics planning, the following requirements (goals) were derived.

- REALISTIC
  - Characteristics definition for data elements used by the optimization algorithms must be defined
  - A Graphical User Interface (GUI) should allow for data elements to be assigned values related to a baseline logistics plan.
  - The GUI should provide for the display of metrics and prove that the optimization algorithms implementations are 1) appropriate to naval logistics planning and 2) are an

advancement on current optimization algorithms for naval logistics planning.

- USEFUL
    - The GUI should be intuitive and useful for a logistics planner
        - Mapping displays should be useful to the logistics community
        - Icons for the standard logistics assets should be understandable
        - Asset types should be reasonably complete.
    - The GUI should suggest alternate optimization strategies to generate a more optimal logistics plan [timeliness, resupply success, risk reduction under adversarial logistics planning conditions]
    - The GUI should display plan simulation results metrics in a format that is comprehendible by the logistics community.

Verification is just a part of a general software simulation assessment. The other parts are validation and accreditation. Accreditation is not discussed in this paper. Validation is indirectly discussed in this paper. Composer 1.0 is a simulator designed to provide optimization algorithms for naval logistics planning. The Composer GUI is provided to verify that logistics plans can be generated and optimized. Accreditation is a process during which a simulator is assessed depending on specified requirements. The Composer 1.0 requirements are applied research and derived requirements for a prototype research simulator. Since the goal of the GUI is to verify and validate algorithms there is no suite of requirements specifications and use cases to provide a pass or fail criteria for simulator accreditation.

This paper is an applied paper and not a research paper. Also, the simulator system under test is partially a black box verification. Stewart Robinson's 1997 paper (Robinson, 1997) discusses confidence and verification (also validation). Verification (also validation) is the key to user's confidence in very complex real-world simulations results. It also discusses block box verification and validation as an answer to the question "does it result in a believable approximation to a real-world problem." Stewart's paper contains multiple discussions including Sargent's 1992 paper (Sargent, 1992). The verification process discussed in this paper is targeted to a naval logistics planning simulator designed to provide optimization techniques for real world naval logistics. The Composer simulator has interfaces for inputs and metrics as outputs.

1.1 Operation of the simulator system under test

As noted above, logistics is complex and the Composer simulator implements the deployment of friendly ship, adversarial assets and friendly logistics ship assets. Composer 1.0 also implements a suite of machine learning optimization algorithms to determine the optimal blue and logistics forces asset placement, commodity loadouts and timelines that result in the optimal execution of a mission under possible adversarial conditions due to the placement on the simulator map of adversarial ship, air and ground assets. The Machine Learning component of the optimization is a hybrid of Bayesian, Neural Network and Clustering. This paper does not discuss validation directly of the Machine Learning component since the Machine Learning component is proprietary. Therefore, extensive validation experiments cannot be developed.

This paper uses certain phrases. One phrase is "asset." Metron's optimization algorithms consider ship, air, facility, commodities and support units as agents. The documentation and this paper uses the phrase asset but the algorithms are operating upon mathematical agents.

The Composer 1.0 simulator is a simulation system that uses scenarios as the configuration to a game engine that is an optimization engine developed by Metron. Simulation runs are called turns. The Simulator interface provides multiple GUI's that allow for the setup of friendly forces, adversarial forces and friendly logistics forces. The main scenario GUI's are the Turn View and the Plan View. These GUI's can display other GUI's that facilitate placement of assets on the simulator map. The user creates scenarios and then uses a "main" menu in the turn GUI to analyze the constructed scenario based upon Metron's optimization algorithms. The simulator can display the analysis of created scenarios and the associated logistics plans graphically. The displayed optimization results are then used to devise a better plan to satisfy a stated mission.

The Composer 1.0 GUI provides for a method to define and position US force assets. Figure 1 illustrates a subset of the types of logistics assets and a view of an asset placed upon the simulator map.

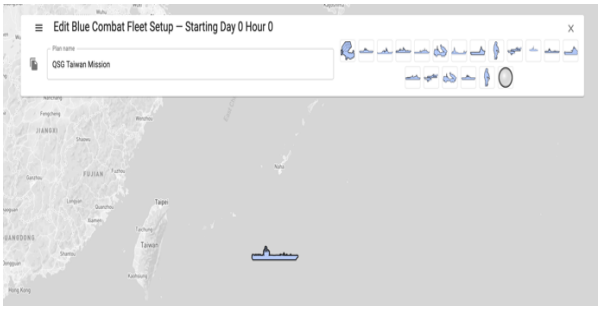

Figure 1. Asset placement for an independent verification of the Composer simulator.

Since this is a naval logistics planning simulator the assets are related to a subset of ship and ground classes. The assets are partitioned into friendly ship and ground classes, adversarial ship and ground classes and friendly logistics ship classes.

Examples of the friendly ship classes are DDG's FFG's LSD's, LSV's LUSV's, MUSV's CVN's, OSV's, LPD's and LSD's. There are a variety of friendly air defense and combat units. Examples of the friendly logistics assets are T-AO's, AS's, T-AKE's, OTS's, T-AOL's, T-AOT's, T-ESB's, T-EPF's and T-AKM's. There are also worldwide ports.

Adversarial assets are related to a subset of ship, air and ground classes. Examples of the adversarial ship classes are FFG's, DDG's, SS's (submarines), SSN's LPD's, LHA's and AGOS. There are also adversarial naval ports and civilian ports. Adversarial assets also include various air assets such as airfields, air defense units, OTH-R radar sites and Satellite assets (SAR/EO/IR).

The Composer 1.0 simulation optimization algorithms can of course work on ships that are sitting still and not performing normal naval operations but that is a trivial goal for a simulator. The idea is to have the blue forces engage in movement and perform various types of activities such as patrolling, air defense and other such activities.

Scenarios are created using the Composer plan view after the turn time duration has been set. The logistics planner assigns assets to the logistics plan by using the various Composer GUI's to perform the following.

- Place the assets on the simulator map.
- Assign values to commodities possessed by the asset. Commodities are the items that will be used by the assets and will be replenished. Examples would be fuel, dry storage and aircraft along with landing craft. Note that no personnel commodities are simulated.
- Generate a set of regions within which to conduct missions and travel routes.
- Assign mission orders to friendly, logistics and adversarial assets. There is a significant suite of mission orders but the primary orders relate to travel, patrolling and transfer of commodities. The travel, patrolling and transfer orders are time-based.

Figure 2 illustrates the scenario used for independent verification.

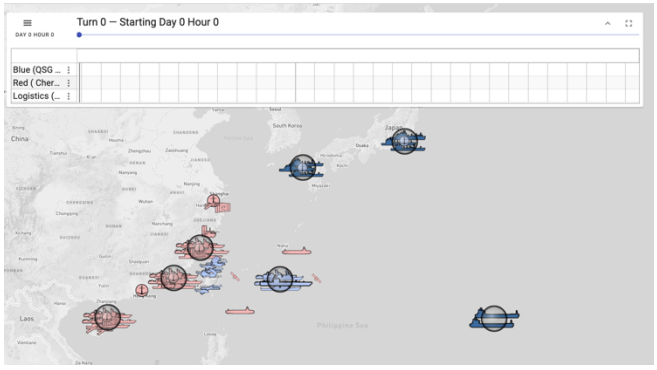

Figure 2. Starting configuration of assets for verification of GUI's and optimizations

Figure 3 illustrates the final configuration of the scenario after the simulation turn has completed.

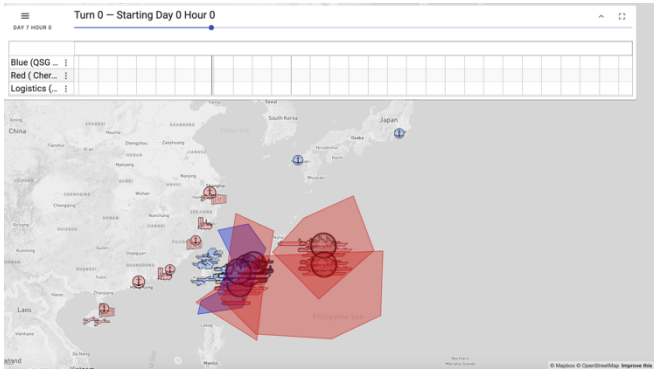

Figure 3. Ending configuration of the simulation run.

The Composer simulator map is an Openstreetmap-based map. Openstreetmap is an opensource mapping and geospatial system (OSMF, 2021). Composer leverages this open-source system to provide the Composer 1.0 simulator interface. The scenario data is placed into a MongoDB open-source database (MongoDB, Inc., 2021). The orders, plans and assets are operated on using XML, JSON objects within a Java language environment. The assets are associated with agents. Since the turns are adversarial this is an example of stochastic optimization.

The simulator run (turn) processes orders, determines commodity usage and resupply. The optimization goal of the simulator is to determine the risk to friendly logistics assets as time progresses dependent upon the movement and capabilities of adversarial assets. The results of the simulator turn are then analyzed and the results displayed in a series of GUI's using various graphical chart types that are color-coded. The results are related to the following

metrics.

- Is an asset disabled due to logistics issues and the time interval of disability?
- What is the risk of a logistics asset being disabled due to adversarial events?
- Hours of asset availability

Metrics are displayed based upon asset or by commodity. Metrics are color-coded.

The goal of the optimization is to rerun turns and compare adjustments to a scenario plan based upon the displayed metrics. The adjustments can be manual or adjustments suggested by the optimization algorithms. Some of the suggested adjustments can also recommend prepositioning of logistics assets, and recommend addition or subtraction of logistics assets.

Essentially the simulator can help design experiments with suggested logistics plan modifications and analyze the results.

1.2 Verification process

Verification and Validation (V&V) are performed by running scenarios and determining the performance of the optimization that results in alternative manually and automatically suggested logistics plans. This *paper focuses on the Verification portion of V&V.*

It should be noted that there are some limitations on the commodities. Aircraft types are a subset of possible military aircraft types and are limited to the major types with no type variants. Fuel and dry storage are bulk values with only 2 types for non-logistics assets. Logistics assets have two to six liquid and dry types of commodities. The optimization algorithms assume various default values for the assets as they optimize the agents assigned to assets.

It should also be noted that initial values in many optimizations can be problematic. The prime example for this V&V process is a very unrealistic plan created for a scenario as a baseline plan.

The Composer 1.0 scenario was delivered by Metron with a set of default scenarios and test scenarios located geographically in the Indian Ocean. The default and test scenarios were executed by Metron. The simulator turn metrics were calculated and alternate manual and suggested plans were executed by the simulator. The results of the optimization algorithms were compared and metrics generated that indicated that the suggested plan adjustments were optimizing.

The verification process generated a separate independent scenario with a baseline plan in Southeast Asia. The verification process also included basic verification that the simulator could be installed.

The composer simulator has a model algorithm that associates risk of being disabled with the probability of detection. In all the scenarios and the independent scenario plans there is an Over the Horizon (OTH-R) asset assigned to the adversarial assets. During the independent verification process the following evaluation steps were performed.

1) Create a 28-day scenario baseline with deliberately included flaws to result in poor planning metrics with multiple asset mission failures.
2) Use the Metrics GUI's to adjust logistics plans.
3) Repeat step 2.

1.3 Results

The following figures illustrate the results of the adjustments of the optimization process. The Metron-supplied suggested adjustments were verified by demonstration which is on method of verification and to a lesser extent validation. Demonstration of various scenarios and following suggested adjustments during a demonstration indicated that the optimization algorithms suggested plan adjustments did produced a series of metrics that were successively more favorable (Metron Inc., 2020).

The rest of this section is focused on the independent verification process results. Figures 4a-i illustrate the refinement and improvement of the logistics planning process via Composer simulation runs after a sequence of plan adjustments. RAS is an abbreviation for Resupply at Station. It equals the days off station since the logistics resupply ships never return to their originating station. The Days Below safety threshold metric is related to the adversarial nature of the simulation scenario. Minimization of this value is an indirect validation of the optimization algorithms since there are OTH-R radars which are inputs to risk calculation of the optimization algorithms. This is not a complete validation of the optimization algorithms. The Metron demonstration of algorithm performance also included demonstrations of the prepositioning and automatic generation of optimized logistics plans. During the independent verification process with the baseline scenario, prepositioning and automatic generation of plans did not complete in a reasonable timeframe. This is suspected due to the assets included in the baseline independent verification scenario. However, the metrics and the steps to adjust plans in the Composer GUI produced no noticeable delay.

Figure 4a is the original independent verification plan with flaws included. The Composer simulator GUI has an automatic asset adjustment tool but the simplest method of adjusting the plan was to manually add a T-AKE dry commodities ship. Figure 4b was the resulting improvement after the first attempt to

resupply ground assets. Figures 4c and 4d illustrate the Composer GUI's for analyzing manual adjustments to logistics plans. Figures 4e, 4f and 4g are metrics calculated by the simulator algorithms after a sequence of adjustments to transfer orders for dry goods and fuel from the added T-AKE asset. Figure 4h illustrates the final supply needs for the ground assets. Adjusting the transfer of supplies based upon 4h the final summary of logistics metrics is illustrated in Figure 4i. Figures 4a through 4i illustrate the improved supply metrics calculated by the Composer simulator. Figures 4a through 4i also illustrate the minimization of the Risk (days below safety threshold) calculated by the optimization algorithms.

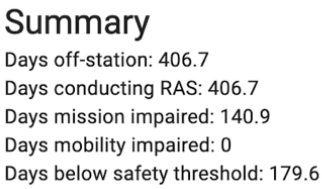

Figure 4a

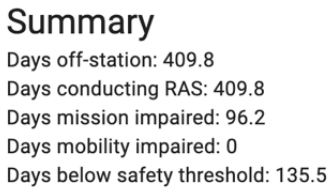

Figure 4b

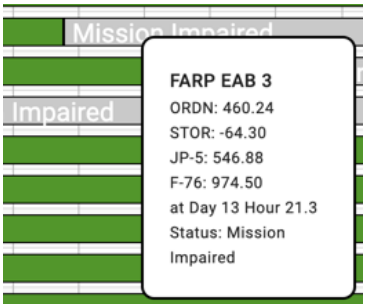

Figure 4c

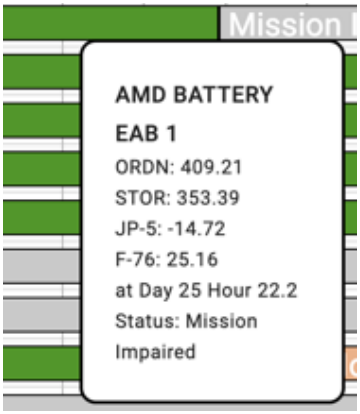

Figure 4d.

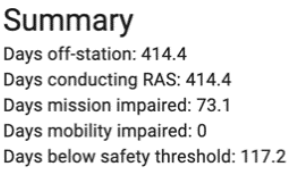

Figure 4e.

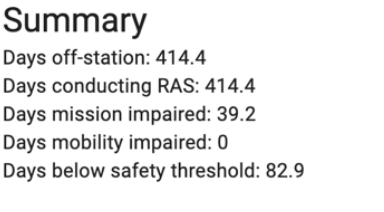

Figure 4f.

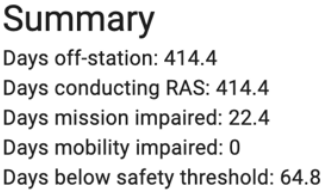

Figure 4g.

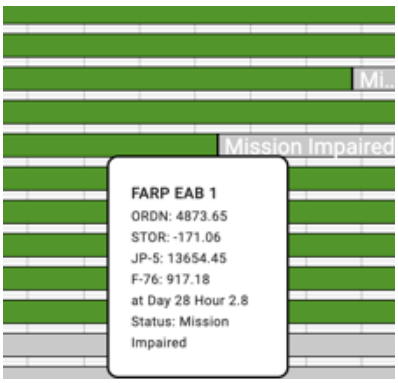

Figure 4g.

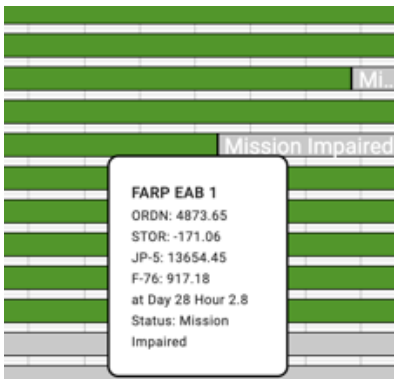

Figure 4h.

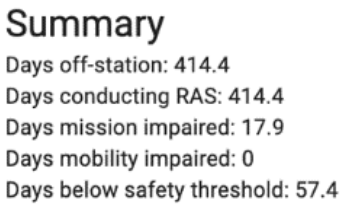

Figure 4i.

Figures 4a through 4i are derived from metric GUIs that have much more extensive timeline display capabilities that can be used for logistics plan optimization. In addition, Figure 3 above illustrates the extreme adversarial nature of the independent verification scenario. The polygons illustrate the geographical proximity of friendly and adversarial assets. The Metron demonstration scenarios had a lower level of adversarial interaction, which may explain the failure of some of the optimization functionality to complete in a reasonable timeframe.

The illustrated results tend to support that verification of the Composer 1.0 naval logistics simulator can support the derived requirements in section 1. In addition, the illustrated results about the days below safety threshold indirectly supports verification and validation of the optimization research requirement.

Figures 5 and 6 illustrate indirect verification and validation of optimization related to logistics planning. Figure 5 illustrates the Generate Logistics Plan process. Although the plan generation as mentioned above could not be verified by demonstration, the Risk Tolerance slider is an input to the heat map generation. Heat map generation is a demonstration method for verifying optimization results. The Risk Tolerance slider in Figure 5 when set to "Low" results in the heat map illustrated in Figure 6. Medium and high-risk Tolerance results in no significant heat map metrics.

Figure 6 below illustrates the result of the optimization related to risk. It should be noted that from the figure 3 polygons related to patrol regions and OTH-R locations that there are significant risks associated with the final successive optimization of the baseline logistics plan.

Figure 5. Risk Tolerance Input parameter GUI element.

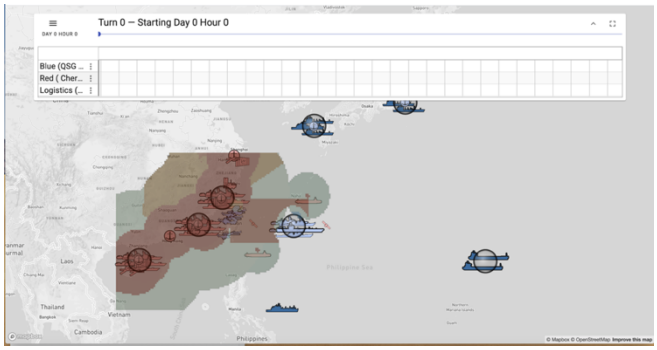

Figure 6. Heat Map result of the Optimization algorithm.

1.4 Conclusion

The Independent execution of the simulator with an independent scenario verified that metrics were displayed to the user by the simulator. Scenario creation was tested and worked as expected given that the simulator was delivered with only basic documentation on usage. The independent baseline scenario with intentional flaws was executed manually producing metrics. The independent scenario was

adjusted and optimized metrics were generated that verified that more successful metrics was generated. The Independent verification scenario also was used to generate a Composer 1.0 simulator user manual (Richardson, 2021).

Since the Independent scenario verification process did not reveal the exact algorithm flow and absolute performance more focused scenarios and inspection of the delivered code are needed for a complete verification of the Composer 1.0 naval logistics planning simulator. Also, the verification process did not include verification of bounds checking against real world shipping logistics data.

1.5 Future work

The Composer 1.0 naval logistics simulator was developed as a DARPA project related to optimization research. Related DARPA sponsored logistics simulators were delivered for Ground logistics. There is also a supply chain logistics simulator utilizing stochastic optimization techniques but also applying optimization to asset commodities with large numbers of commercially validated commodity characteristics. Future work is planned related to comparing the algorithm performance of these other simulators.

References

Metron incorporated. (2020). Composer 1.0.1 user guide.

Mongo DB, Inc. www.mongodb.org. Retrieved April 15, 2021.

OpenStreetMapFoundstion(OSMF). www.opensstreetmap.org. Retrieved April 15, 2021.

Richardson, J. (2021) Composer Quick start guide. *Naval information warfare center*.

Robinson, S. (1997). Simulation model verification and validation: increasing the user's confidence. *Proceedings of the 1997 winter simulation conference*.

Sargent, R. G. (1992). Validation and verification of simulation models. *Proceedings of the 1992 winter simulation conference*.

Funding

This research was funded by the Defense Advanced Research Project Agency. Project DARPA funding numbering is internal to the Naval Information Warfare Center.