Agent Based Simulation of Naval Tactics with Effectiveness Analysis Features

Erkin Çilden, Ahmet Sezer, M. Haluk Canberi
STM Defense Technologies Engineering and Trade Inc.
Ankara, Turkey
{erkin.cilden, asezer, hcanberi}@stm.com.tr

ABSTRACT

Developing optimal naval tactics is a complex problem domain by definition. A common practice is to simulate the process with possible input variations and use the outcome of its repetitive execution as a means for decision support. One of the challenges is that, for various reasons, stakeholders tend to agree on a fixed level of fidelity and scale, which is very difficult to alter at the later stages of the product life-cycle. Another challenge is the missing integrated effectiveness analysis tools, since the post-experimentation phase is usually presumed to follow a regular data-analysis process. This study proposes an agent based approach to simulation of naval tactics, with integrated effectiveness analysis features.

The agent based design incorporates multiple levels of modeling abstractions. The first level is the computation primitives of the models. Behavior Trees are used for the second level, with expressive power similar to DEVS (Discrete Event System Specification) and FSM (Finite State Machine). The agency abstraction is the third level, partitioning the problem space into two: the agents with bounded rationality, the environments with which the agents interact. Last level is the closed loop scenario definition required to run the simulation, by using the events fired and consumed by the agents.

The second challenge is attacked via a data centric approach to integrate effectiveness analysis features. The user is guided through analysis steps: selection of performance measures from the output logs of a specific simulation experiment and assigning their interpretation methods, construction of an effectiveness tree using a subset of the performance measures as leaf nodes, defining intermediate nodes to aggregate with predefined methods (including MCDM , Multiple Criteria Decision Making) and user defined methods, and evaluation of the result using successive queries on the database.

The paper explains how to integrate multiple levels in proposed agent based design, and discusses the integrated tree based effectiveness inference mechanism, for naval tactics simulation.

ABOUT THE AUTHORS

Erkin Çilden has a Ph.D. in Computer Engineering from Middle East Technical University, and currently is a group leader and senior software engineer at STM. His research interests include multi-agent systems, reinforcement learning and simulation software design.

Ahmet Sezer has a B.S. and M.S. in Mechanical Engineering from Middle East Technical University, and currently is a group leader and senior systems engineer in STM. His expertise is on simulation modeling (weapon systems, flight dynamics) and accreditation.

M. Haluk Canberi has a B.S. in Electrical and Electronics Engineering from Hacettepe University, and an MBA from Hacettepe University. He is a group manager and senior systems engineer at STM. His expertise includes avionics, simulation systems.

Agent Based Simulation of Naval Tactics with Effectiveness Analysis Features

Erkin Çilden, Ahmet Sezer, M. Haluk Canberi
STM Defense Technologies Engineering and Trade Inc.
Ankara, Turkey
{erkin.cilden, asezer, hcanberi}@stm.com.tr

INTRODUCTION

A known challenge in requirements management of military simulation projects is governing the changes in model fidelity levels. Although the Model Hierarchy of Military Simulation (Tolk, 2012) provides a common outline to agree on, specific requirements of a project may impose variety on fidelity levels of models within the same model space. Due to the complex nature of the naval tactics, especially self-engaging sub-systems are modeled in engineering level, while maneuvering models are of a lower level of fidelity, which means different levels in the Model Hierarchy should co-exist, and possibly interact.

Such requirement changes usually stem from either computational/performance restrictions, or the need for flexibility to locally switch the model hierarchy according to the intended use. Most of the time, the intended uses of the project are either slightly changed, or extended throughout the project life-cycle. Although the fidelity variation might be quite reasonable in the customer's point of view, they usually raise challenging software engineering problems, difficult validation and verification (V&V) questions, and tricky model development issues. The accumulation of these changes may result in adverse effects on the maturity of the product at the end of the project.

The first part of this paper focuses incorporation of a novel pattern based on agent based design principles, in order to achieve simulation models that are relatively immune to changes of model fidelity requirements, specifically for analysis simulations for naval tactics.

On the other hand, military executives and commanders started taking increasing interest in getting scientific answers to questions pertaining to system acquisition, threat perception and quantification, assessment of damage or casualties, evaluation of chance of winning a battle, types of systems need to be deployed to provide cost-effective defense. Decision makers must be able to take rapidly changing information and technologies and combine them with projected situations in order to make decisions and understand their consequences (Jaiswal, 1997). What these decision makers are looking for is a quantification of system effectiveness for complex warfare systems.

Evaluating the effectiveness of a complex system has been an increasingly challenging problem (Habayeb, 1983). In a military environment, system effectiveness usually has a special meaning. In the classical approach, military platform design was characterized on optimum performance. Recently, the paradigm has shifted the emphasis from optimum system performance to overall system effectiveness as a key measure of merit. Today, however, system effectiveness should instead focus on mission success, given a number of threats and situations, taking the life cycle costs into account (Soban & Mavris, 2001). Although there is an apparent need for integrated effectiveness analysis tools for tactical simulation software, to the authors' best knowledge, the literature is missing generally applicable design patterns.

The second contribution of this paper is a data centric approach to integration of effectiveness analysis features to naval tactics analysis simulations. Although the specific case studies address the naval tactics simulations, the ideas are generic and can be applied to other military domains.

RELATED WORK

Agent-Based Simulation

Since the early studies of multi-agent systems, agent-based design has been proven to be a useful tool in many disciplines. Likewise, it also changed the way of modeling real-life phenomena, making a shift from previously dominating paradigms, namely system dynamics and discrete event based modeling, to the multi-agent based simulation to observe the emergent behavior of systems (Siebers et al., 2010). The key principle that distinguishes the agent based approach from others is the notion of *autonomy*. Usually, agent based modeling is coupled with the multi-agent system design (Weiss, 2013), since it is essentially empowered by the ability of multiple agents to yield an emergent behavior for a complex phenomenon.

Agent-based simulation is based on the very same principles. As the complexity of the simulated processes increases, the top-down modeling approaches become infeasible. It is more convenient instead to model the complex systems in a ground-up fashion, so that the emergent behavior of the multi-agent system informs the dynamics of the overall system (Macal & North, 2008). Although the main focus of agent-based simulation has been the disciplines such as biology, ecology, economics, sociology, physics, chemistry and networked systems (Xiao et al., 2018), where the agent design is relatively straightforward due to the nature of the problems, the concept can easily be adapted to problems where the processes are inherently complex, such as naval warfare tactics. The main agent based design principles are almost the same, and similar frameworks can be incorporated as development platforms and scheduling engines (Cil & Mala, 2010; Cioppa et al., 2004; Railsback et al., 2006).

Simulation of Naval Tactics

Naval tactics consist of various elements like the operation of subsystems, maneuvering, communication, and command-control. War games and analysis simulations -also called analytical games- (Caffrey, 2019) have been used for different needs. War gaming simulations allowed users tactical trials and also offered personnel training. Execution of military tactical scenarios via simulation tools has gained attention for more than a few decades, since the tactics are complex in nature and can be effectively modeled due to well-established rules executing the sub-systems and decision making regulations (Jeong et al., 2017).

Since analysis simulations focus on essentials of the process, repeated runs and scenario variation mechanisms such as stochasticity, they are typically executed in a closed-loop fashion (without user intervention during execution) and with speeds at best-effort. For exploration of best tactics, experimentation and multiple runs are carried out to support the tactical decisions in a controlled way (Seo et al., 2011).

Model Fidelity Requirements

For naval tactics simulations, as for military simulations in general, model fidelity requirements are usually either unclear at the beginning of a simulation project, or are prone to change during development. Moreover, it might be the case that models of different fidelities co-exist within the same model space, although studies exist on interoperation of simulations of different abstraction levels (Hong et al., 2011).

The systematic ways of coping with changing requirements, together with other essential process steps, has been argued (Balci, 2012). However, there is another aspect of the problem concerning the software design aspect of model requirements, which is usually attacked via distributed simulation principles (HLA, DIS etc.) and reusable model software (Taylor, 2019).

Behavior Modeling

Especially in agent based simulation context, it is a common practice to use stateful representations of behaviors to define and reuse the activities of the model as a function of time. There are various methods that the model designer may benefit, few of which became popular due to their expressive power and ease of use. Finite state machine based methods (FSM, DEVS), and tree based methods (decision trees and behavior trees) show up as most common approaches (Colledanchise & Ögren, 2018).

While the main advantages of finite state machine based methods are their wider usage in computer science, being easy to understand and implement, they are prone to scalability and reusability issues. Tree based methods, on the other hand, take advantage of modularity, intuitive structure and hierarchy. However, decision trees suffer from the lack of proper failure handling mechanism. Behavior trees, initiating from the game industry, are known to be human readable, modular, hierarchical and reusable. On the other hand, checking all the conditions can be expensive and implementation can be complex for behavior trees.

In naval tactic simulations, a decision to model behaviors should take into account different aspects of the project such as stakeholders' profiles (developer, analyst etc.), reuse frequency of different behaviors, changes in existing behaviors, complexity of simulation scenarios, time spent of scenario planning and number of simulation entities.

Effectiveness Analysis
The effectiveness of a system can be defined as the degree of accomplishing its goals within a defined task. The effectiveness of a naval platform can be described as the rate of successful fulfillment of warfare tasks. There are many examples in the literature on evaluating the warfare effectiveness of different platforms. The initial studies on this subject calculate effectiveness measures based on firepower. However, the effectiveness of a ship is affected by many factors apart from its firepower. These factors include constructs such as abilities, environment, survivability, reliability, endurance, and availability (Roedler & Jones, 2005; Unlu, 2015). Also, the soft metrics such as the cost of a system should also be taken into account (Kitowski, 1992).

In the evaluation of the integrated effectiveness with the combination of the effects of many factors, the evaluation of the effectiveness indices can be done by using the multi-criteria decision-making (MCDM) techniques (Dong et al., 2008b, 2008a). There are MCDM related studies, such as effectiveness measures evaluation by using multiple MCDM techniques together (Shao & Chen, 2014; Wang et al., 2008) and probability-based effectiveness value calculations (Seo et al., 2011; Zhao et al., 2017). Studies also exist for warfare effectiveness analysis by using mathematical modeling, optimization and simulation techniques (Przemieniecki, 2000; Tolk, 2012).

MANAGING CHANGES IN MODEL FIDELITY REQUIREMENTS

Levels of Model Abstractions
As a common pattern in all engineering disciplines, the key instrument to scale-up the solutions of smaller problems to be applied in larger problems is the *abstraction* mechanism. Usually coupled with the well-known *divide-and-conquer* principle, abstractions make a specific solution practically reusable and interoperable.

Naval tactic simulation requirements usually stem from the analysis of sub-system reaction performances. However, gathering the outputs of performance measures for sensors, countermeasure systems, weapon systems, etc. is expensive and time consuming most of the time. As addressed in the Model Hierarchy of Military Simulation, the output of the lower level should be the input if the upper level, thus, engineering level simulation tools are used to generate the required data to be used in an upper level. Based on the agent directed design principles, the proposed architecture has four levels of abstraction, each with its own aspect of contribution for immunity to requirement changes (Figure 1).

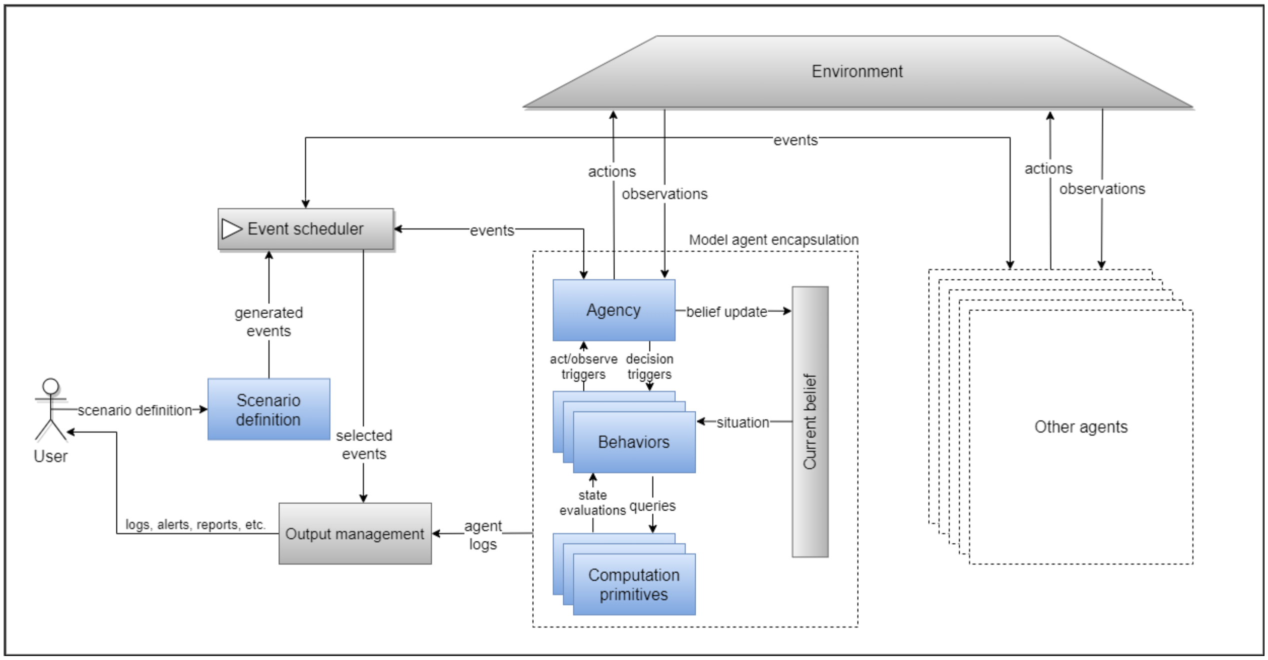


Figure 1. Roles of proposed modeling abstractions (colored in blue) within the agent based simulation design

Computation primitives

This is the level where the execution semantic mimics the engineering calculations of the real system which is encapsulated by the simulation agent. As an example, consider a scenario containing a single ship model, defending itself against a single surface to surface missile (SSM) model. Ship model has a child missile launcher model with a single surface to air missile (SAM). SSM model is placed to the south-east of the ship initially, for a suitable engagement.

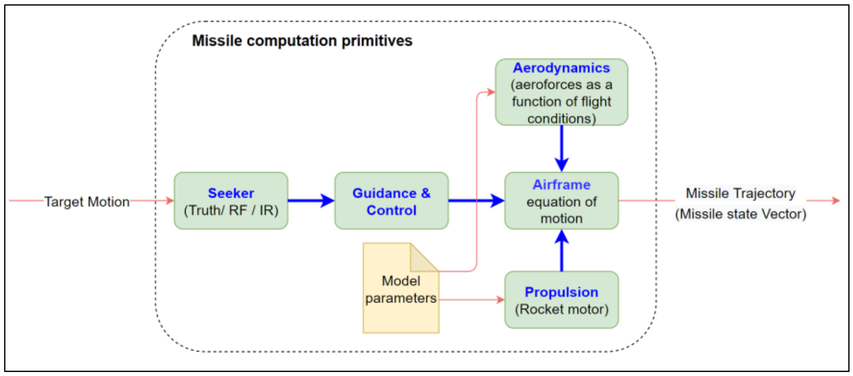


Figure 2. Computation primitives of a missile model

SSM is constituted on missile core model where necessary computation primitives are available to construct behaviors of SSM model (Figure 2).

Although internals of the computation primitives of SSM are not essential for the purposes of this paper, as two examples, guidance and control functionality interprets target relative motion and calculates commanded acceleration according to proportional navigation laws (Faruqi, 2012), and airframe combines acceleration commands, aerodynamic and propulsion forces together with missile instant state vector and utilizes a numeric solver to calculate missile state vector for next time step (Brochu & Lestage, 2003).

Behavior level makes queries to this level for state change conditions and for controlling the execution continuity. Obviously, the computation primitives may also provide interfaces (via function interfaces, services, etc. depending on the software architecture) to be used at the upper level, if required. Since computation primitives are isolated, it is relatively easy to cope with requirement changes concerning engineering level alteration or calibration requests.

Behaviors

This level is where the model state changes are controlled. Behavior Trees (Colledanchise & Ögren, 2018) provide a powerful means to define and control state changes, as well as action decisions. This level makes use of the agent's belief on the current environment conditions, and possible triggers coming from agent's higher level decision logic.

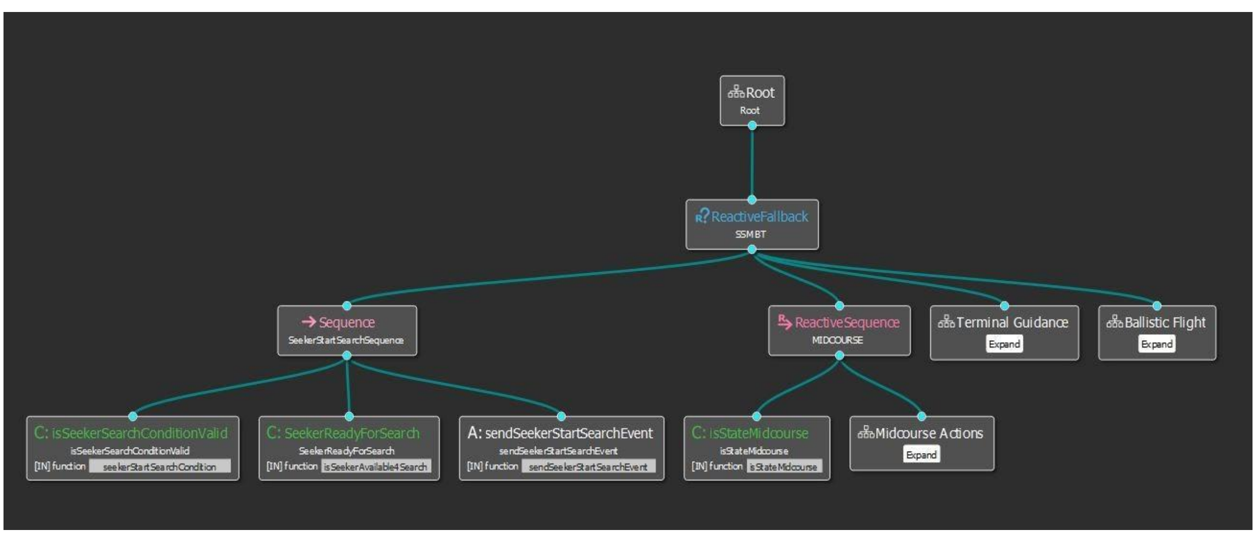


Figure 3. First 4 levels of nodes of SSM behavior tree

Following the SSM case study, the missile model may go through midcourse, terminal guidance and ballistic flight states during the simulation. At this stage state query and state trigger functions allow to construct and manage model behavior. Behavior tree representation is chosen for SSM model. Main atomic/core functions were extracted to the user as action and condition nodes to construct fallback and sequence controls of a general behavior tree. Main conditions nodes were generally selected as internal state identifiers, whereas action nodes were selected as state triggers or action handlers.

Although it is perfectly fine to use earlier state based formalisms such as DEVS (Zeigler, 1976), finite state machines, decision trees etc., it is much easier to build up complex decision mechanisms, reuse the design and make changes with a tree data structure. Hence, when a model's state management is handled by using Behavior Trees, it is much easier to handle requirement changes concerning the state transition logic, since the state management design is isolated from computation primitives.

Agency

Although behavior level might be enough to model relatively simple and non-complex systems, a regulating agency mechanism is required to orchestrate the simulation scene, and possibly higher levels of rationality.

The key construct for agency abstraction is the environment design, with which all the agents interact individually. While the environment represents the current snapshot of the simulation scene, the event scheduler regulates the time axis of the simulation. Note that the agent both updates the environment via its actions, and affects other agents via the event mechanism. Our implementation also allows the agents to advance time without event triggers, but this functionality is not required if the simulation is solely event driven.

Agency level is where model-environment interaction realism is implemented. With this notion of agency, the higher level decision logic is encapsulated and is flexible to be re-designed throughout a project. Depending on the requirements and the nature of the system, this level can be implemented as a reactive agent, a BDI agent (Rao & Georgeff, 1995), or even a learning agent. Obviously, the agency level would need to keep track of his belief on the current state of the environment, but belief tracking may not be required for simpler agents, such as reactive agents. In our implementation of naval tactics, all agents except Command and Control model agents are implemented as reactive agents. Command and Control agent is more like a BDI agent where it keeps track of its belief in the form of common tactical picture extended with commander goals.

Back to the SSM example, interaction among missile and other models, including child models, are executed by event mechanisms and environment models. Missile model is related with physical and RF environment models in this study, as the model has a footprint on the related aspect of the simulation scene (position, radar cross section values, velocity, orientation, etc.). Model agent basically reads/writes on the physical environment and listens to events that are registered.

Finally, it is important to note that an agent abstraction can make use of more than one behavior (and computation primitives) abstractions. For example, for a missile agent, a possible design practice can be to partition the behaviors into three, fuse, seeker and body, for each of which a different behavior tree can be implemented, but the final model is aggregated at the agency level.

Scenario Definition

Events are the primitives of the scenario definition abstraction level. Scenario definition step essentially compiles user defined scenario into a set of conditional events. Following the definition of condition-event pair list by the user, a dependency and consistency control is executed. After this process, the generated condition-event pair list is fed into the event scheduler during run-time, in a controlled fashion, starting with the events to be triggered at the very beginning of the simulation execution. In a closed loop simulation execution, there is no user intervention during the simulation run, but the events are injected into the loop whenever the corresponding condition holds.

Scenario definition abstraction can be as simple as simulation time-event pairs, where the manager triggers the corresponding event at the given time. However, time is usually not enough as a condition, and more sophisticated trigger conditions are required. It is possible here to incorporate more formal scenario design formalisms, such as Petri Nets, BPML, etc. depending on the complexity of changing requirements on scenario definition throughout the project.

It is relatively easy to re-design the scenario definition method which fulfils the user's needs, based on the events as the primary constructs as an abstraction layer.

The SSM case study couples with the scenario definition user interface at this point. Assume the scenario involves a ship with a missile launcher, which would defend itself against a selected surface to surface missile. Primary events to be defined to initiate the agent are "ADD" and "INIT" events. Although these events are common for most of the models, there are also model-specific events those could be defined in the scenario editor. For example, "SPAWN" for SSM model and "MISSILE_LAUNCHER_FIRE" for missile launcher model (Figure 4) are defined via scenario manager, which are interpreted as "INIT" events by the simulation engine.

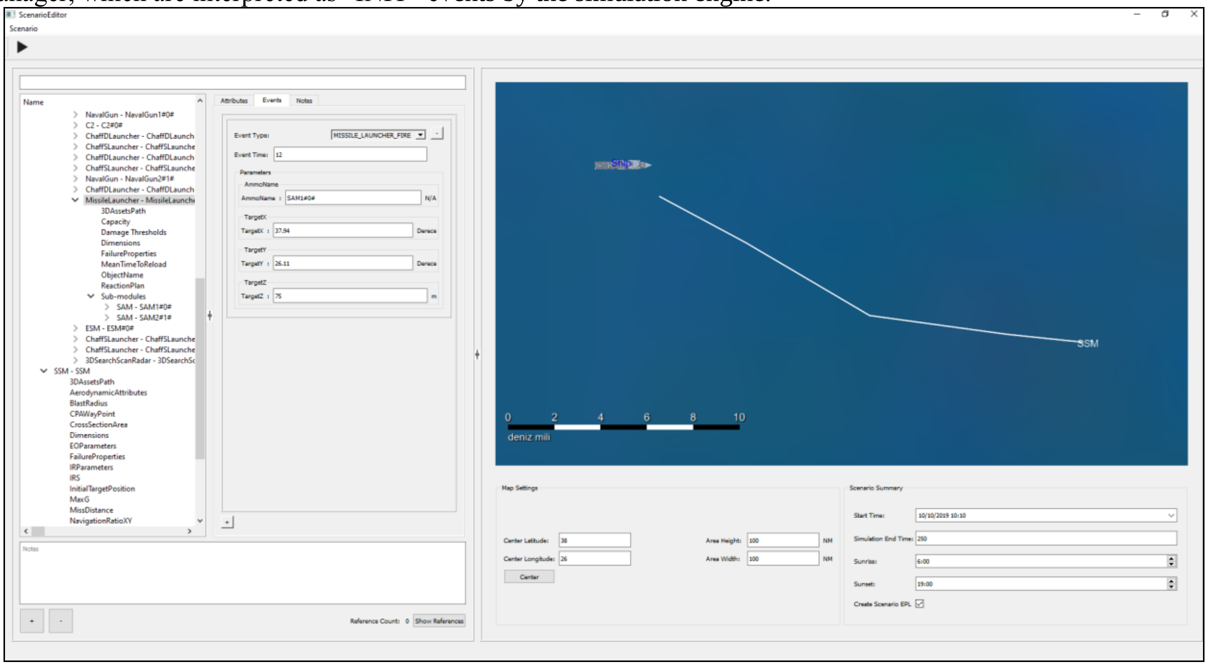


Figure 4 Scenario definition via events

INTEGRATED EFFECTIVENESS ANALYSIS

In the integrated effectiveness analysis model developed, the overall effectiveness of the naval platform was defined by the combination of performance measures and effectiveness measures. An effectiveness tree is created by organizing the defined measures according to their hierarchical structure. As a result of analyzing the created effectiveness tree by using MCDM methods, the integrated effectiveness value of the naval platform is evaluated.

During the effectiveness analysis of a platform, one or more fundamental objectives may need to be evaluated. For this reason, the effectiveness trees can be formed multi fundamental objectives and means objectives that serve the evaluation of the overall effectiveness of a platform (Keeney, 1992). Before diving into the design and implementation details, the reference software architecture is briefly described in the next section to provide a basis to properly understand the integration principles of the proposed effectiveness analysis process.

Reference Architecture

Although the modeling is agent driven, the integrated effectiveness evaluation features are achieved via the data centric design approach. In fact, one can see that the software system architecture is defined by two axes: while the horizontal design axis is inspired by the well-known "pipes-and-filters" pattern (Buschmann, 1996), the vertical axis follows a data-centric approach (Figure 5).

Since the effectiveness analysis requirement imposes an experimentation over a defined simulation scenario, the reference architecture guides the user throughout the control flow (Özkan et al., 2019). Each module can be thought of as a standalone application whose concern is the data store that is assigned for itself, but rules engaged for the underlying database dependencies ensure the integrity among the modules. Thus, the primary asset of this architecture is simulation data.

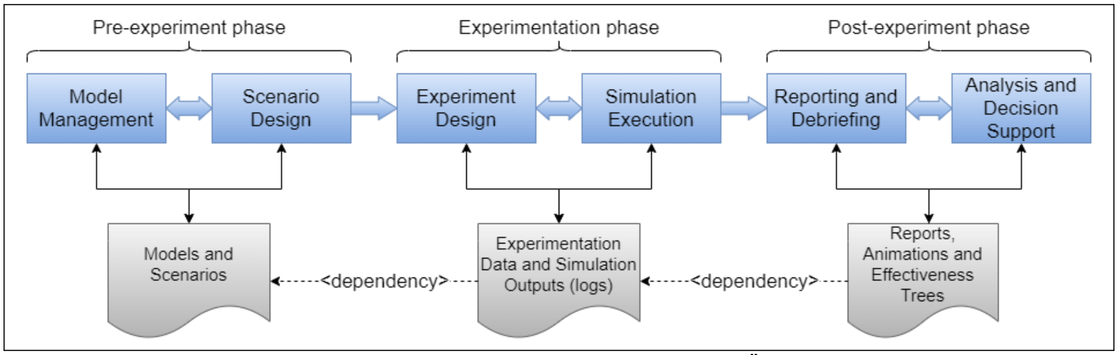


Figure 5. Data centric reference software architecture (Özkan et al., 2019)

The Effectiveness Analysis Flow

The above mentioned architecture provides the data centric means to efficiently integrate the effectiveness analysis features to the naval tactics simulation system. Based on the data flow aligned with this control flow, the integral steps of the effectiveness analysis features are shown in Figure 6.

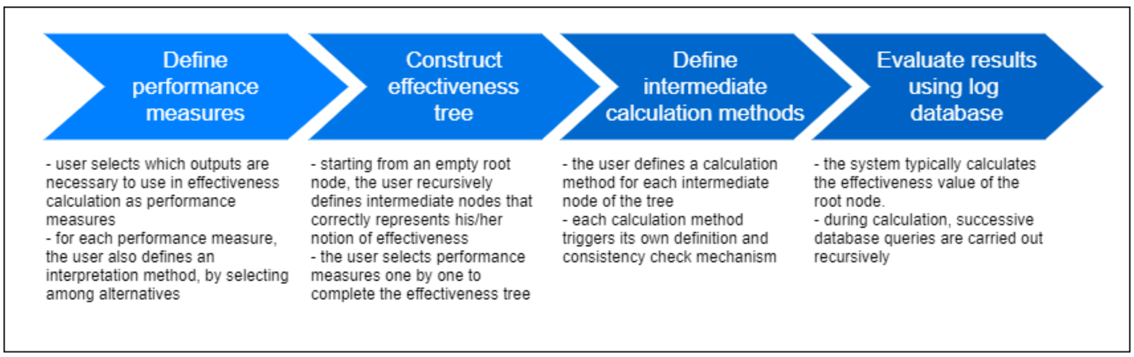


Figure 6. Integrated effectiveness analysis capability

Definition of Performance Measures

The first step of the effectiveness analysis is the selection of performance measures from the output logs of a specific simulation experiment and assignment of their interpretation methods. All experiment outputs are logged into the database, tagged by experiment, scenario instance and trial identifiers. Using the performance measure definition tool, the user is able to select which log items would be included in the effectiveness calculation. Performance measure selection is conducted only among the output logs of the experiment, not among the whole possible performance measure set. At this step, the user can also define external performance measures (i.e. data not produced by the simulation system, but are important for that specific effectiveness analysis), such as cost, weight, etc.

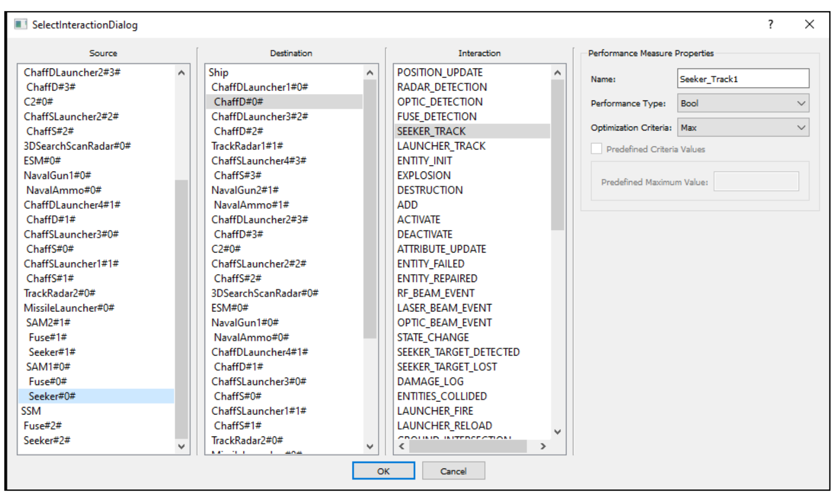


Figure 7. Performance measure definition example

The example in Figure 7 shows how the user defines the “tracking” interaction between seeker and chaff models as a performance measure. The user states that the performance type is Boolean (meaning the seeker is either tracked or not tracked).

Construction of the Effectiveness Tree

Effectiveness tree is essentially an n-ary tree with weighted edges. From the list defined by the user in the previous step, the user is asked to make performance measure selections into the tree. Starting from an empty root node (top of the effectiveness tree), the user recursively adds intermediate (aggregation) nodes, based on the intended effectiveness hierarchy in mind. The performance measures can only be used once, and only at the leaf nodes. At the end, the whole tree consists of performance measures as leaves, and conceptual effectiveness values as intermediate nodes (Figure 8).

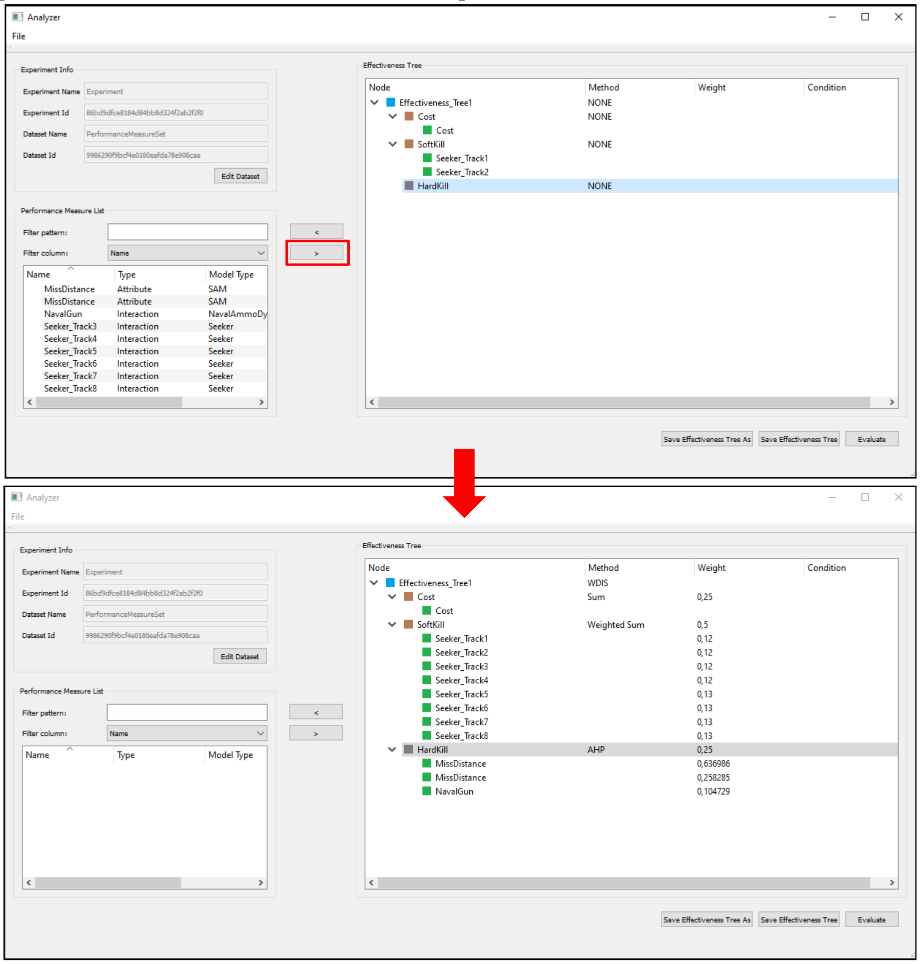


Figure 8. Effectiveness tree construction

By definition, the effectiveness tree should be extended with edge weights and intermediate node calculation methods, so that the evaluation phase will know how to interpret the contributions of each node to its parent as well as how to calculate this contribution. At this point, the user defines the weights of nodes and the calculation method (among a list including AHP, weighted sum, etc.) for every intermediate node. Depending on how sophisticated the method is, the software assists the user to complete the details of the selected method through additional dialog windows, if necessary (Figure 9 is an example of an AHP dialog box).

Evaluation of Effectiveness Values

Finally, after completion of the effectiveness tree with all its required data, the system evaluates the effectiveness values of every node in the tree by means of

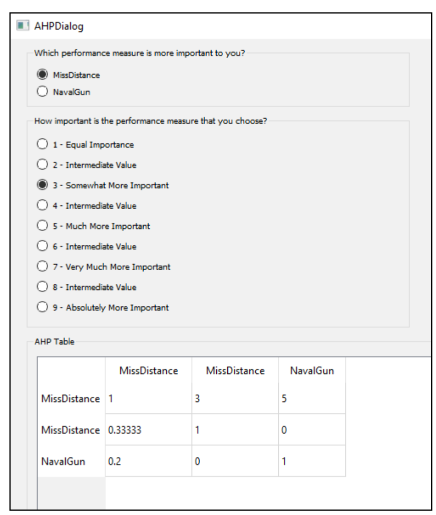


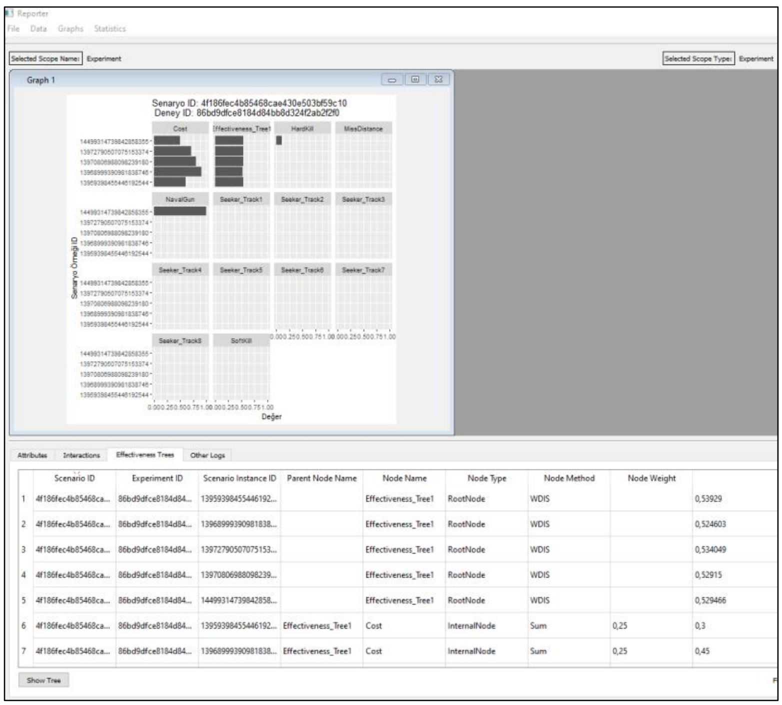
Figure 9. AHP dialog box example

- the leaf nodes (performance measures) by querying and normalizing necessary information among all the experiment results, and applying the selected interpretation method for the corresponding measure,
- user defined weights and methods for the intermediate nodes to calculate their values
- the tree structure to incrementally calculate the root node value

At the end, there is a single effectiveness value, which is obviously not meaningful alone, but is valuable when compared with other instances of the same scenario within an experiment. Moreover, the user can investigate the contribution of intermediate nodes for further analysis, by using integrated or external reporting and analysis tools (Figure 10).

Definition of Calculation Methods

Modernization of sub-systems constitutes a sample decision-making scenario to illustrate how the integrated effectiveness analysis capability makes a difference. Consider a scenario the analyst aims to assess how to use a certain system modernization budget. Assume that the available budget lets him completely replace the existing surveillance radar, and there are two alternatives to buy. What he needs to do is construct a number of base scenarios spanning the minimum tactical requirements, design experiments based on these scenarios to reflect the required system variations, design effectiveness trees representing a subjective evaluation of the simulation outputs, and recursively evaluate and analyze the effectiveness trees. He can also investigate the impact of performances of other systems to the overall effectiveness (such as the impact of ship speed or orientation on radar performance) using impact of sub-tree evaluation results to the higher level nodes of the tree.



	Scenario ID	Experiment ID	Scenario Instance ID	Parent Node Name	Node Name	Node Type	Node Method	Node Weight	
1	4f186fec4b85468ca...	86bd9dfce8184d84...	13959398455446192...		Effectiveness_Tree1	RootNode	WDIS		0,53929
2	4f186fec4b85468ca...	86bd9dfce8184d84...	13968999390981838...		Effectiveness_Tree1	RootNode	WDIS		0,524603
3	4f186fec4b85468ca...	86bd9dfce8184d84...	13972790507075153...		Effectiveness_Tree1	RootNode	WDIS		0,534049
4	4f186fec4b85468ca...	86bd9dfce8184d84...	13970806988098239...		Effectiveness_Tree1	RootNode	WDIS		0,52915
5	4f186fec4b85468ca...	86bd9dfce8184d84...	14499314739842858...		Effectiveness_Tree1	RootNode	WDIS		0,529466
6	4f186fec4b85468ca...	86bd9dfce8184d84...	13959398455446192...	Effectiveness_Tree1	Cost	InternalNode	Sum	0,25	0,3
7	4f186fec4b85468ca...	86bd9dfce8184d84...	13968999390981838...	Effectiveness_Tree1	Cost	InternalNode	Sum	0,25	0,45

Figure 10. Analysis of an evaluated effectiveness tree

As a result, since the analyst can perform the overall sequence forth and back repetitively, it would be easier and time consuming to execute the performance measure selection, experiment design and effectiveness evaluation cycle for a full-fledged analysis of the decision question.

CONCLUSIONS AND FUTURE WORK

The contribution of this study is two-fold: First, an agent based model abstraction approach is presented, in order to cope more easily with changes in model fidelities throughout the project life-cycle. The other contribution is on how to define and integrate effectiveness analysis capabilities to a military analysis simulation in a data centric fashion. Although the proposed methods are evaluated through naval tactics simulation case studies, the ideas are generic and are applicable to different military simulation domains.

An obvious future work is capturing the changes in model fidelities through a number of metrics during the project and interpreting the measurements afterwards. Analyzing the possible side effects (such as computation performance) of the proposed methods is another future study. As another future study, the scenario definition level should be enriched by formal scenario definition methods, which would be a valuable contribution to the model abstraction idea. It might be interesting for the reader to note that, the project is still in progress and the next challenge is modeling of naval task group missions. Therefore, the modeling abstraction approach will be subject to stress testing in the near future.

ACKNOWLEDGEMENTS

This work is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 115A044. The authors would also like to thank Gökçe Özkan and Denizcan Yılmaz for their support to generate images. The authors are also grateful to the GEMED project team for their patience and effort. This work uses the software components contained within the NRL SIMDIS computer package written and developed by the Naval Research Laboratory, Tactical Electronic Warfare Div.

REFERENCES

Balci, O. (2012). A life cycle for modeling and simulation. *SIMULATION*, *88*(7), 870–883. https://doi.org/10.1177/0037549712438469

Brochu, R., & Lestage, R. (2003). Three-degree-of-freedom (DOF) missile trajectory simulation model and comparative study with a high fidelity 6DOF model. *DRDC Valcartier TM*, 2003–2056.

Buschmann, F. (Ed.). (1996). *Pattern-oriented software architecture: A system of patterns*. Wiley.

Caffrey, M. B. (2019). On Wargaming: How Wargames Have Shaped History and How They May Shape the Future. *Newport, RI: Naval War College Press*.

Cil, I., & Mala, M. (2010). A multi-agent architecture for modelling and simulation of small military unit combat in asymmetric warfare. *Expert Systems with Applications*, *37*(2), 1331–1343. https://doi.org/10.1016/j.eswa.2009.06.024

Cioppa, T. M., Lucas, T. W., & Sanchez, S. M. (2004). Military applications of agent-based simulations. *Proceedings of the 2004 Winter Simulation Conference, 2004.*, *1*, 180. https://doi.org/10.1109/WSC.2004.1371314

Colledanchise, M., & Ögren, P. (2018). *Behavior Trees in Robotics and Al: An Introduction*. CRC Press.

Dong, C., Wu, D., & He, J. (2008a). Decision Analysis of Combat Effectiveness Based on Rough Set Neural Network. *2008 Fourth International Conference on Natural Computation*, 227–231. https://doi.org/10.1109/ICNC.2008.508

Dong, C., Wu, D., & He, J. (2008b). Study on Method of Satellite Navigation System Combat Effectiveness Evaluation Based on Rough Fuzzy Sets. *2008 Chinese Control and Decision Conference*, 3220–3223.

Faruqi, F. A. (2012). *Integrated Navigation, Guidance, and Control of Missile Systems: 3-D Dynamic Models* (Technical Report DSTO-TR-2706). Australian Government Department of Defence, Defence Science and Technology Organisation, Weapons Systems Division.

Habayeb, A. R. (1983). *System Effectiveness*. Pergamon Press.

Hong, J. H., Seo, K. M., Seok, M. G., & Kim, T. G. (2011). Interoperation between Engagement-and Engineering-level Models for Effectiveness Analyses. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *8*(3), 143–155. https://doi.org/10.1177/1548512910389483

Jaiswal, N. K. (1997). *Military operations research: Quantitative decision making*. Kluwer Academic Publishers.

Jeong, D.-H., Roh, M.-I., Ham, S.-H., & Lee, C.-Y. (2017). Performance analyses of naval ships based on engineering level of simulation at the initial design stage. *International Journal of Naval Architecture and Ocean Engineering*, *9*(4), 446–459. https://doi.org/10.1016/j.ijnaoe.2016.12.002

Keeney, R. L. (1992). *Value-focused thinking: A path to creative decisionmaking*. Harvard University Press.

Kitowski, J. (1992, February 3). Combat effectiveness methodology as a tool for conceptual fighter design. *Aerospace Design Conference*. Aerospace Design Conference, Irvine,CA,U.S.A. https://doi.org/10.2514/6.1992-1197

Macal, C. M., & North, M. J. (2008). Agent-based modeling and simulation: ABMS examples. *Proceedings of the 40th Conference on Winter Simulation*, 101–112.

Özkan, G., Çilden, E., Gültekin, E., Sezer, A., & Canberi, H. (2019). A reference architecture for military simulations with experimentation and analysis features. *Proceedings of the 2019 Summer Simulation Conference*.

Przemieniecki, J. S. (2000). *Mathematical Methods in Defense Analyses*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/4.862137

Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION*, *82*(9), 609–623. https://doi.org/10.1177/0037549706073695

Rao, A. S., & Georgeff, M. P. (1995). *BDI Agents: From Theory to Practice*. 8.

Roedler, G. J., & Jones, C. (2005). *Technical Measurement. A Collaborative Project of PSM, INCOSE, and Industry:* Defense Technical Information Center. https://doi.org/10.21236/ADA605916

Seo, K.-M., Song, H. S., Kwon, S. J., & Kim, T. G. (2011). Measurement of Effectiveness for an Anti-torpedo Combat System Using a Discrete Event Systems Specification-based Underwater Warfare Simulator. *The Journal of*

Defense Modeling and Simulation: Applications, Methodology, Technology, *8*(3), 157–171. https://doi.org/10.1177/1548512910390245

Shao, X. F., & Chen, Q. (2014). Complication Evaluation of Methods of Naval Battlefield Electromagnetic Environment Based on Combat Effectiveness. *Applied Mechanics and Materials*, *565*, 233–237. https://doi.org/10.4028/www.scientific.net/AMM.565.233

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204–210. https://doi.org/10.1057/jos.2010.14

Soban, D., & Mavris, D. (2001, October 16). The need for a military system effectiveness framework—The system of systems approach. *1st AIAA, Aircraft, Technology Integration, and Operations Forum*. 1st AIAA, Aircraft, Technology Integration, and Operations Forum, Los Angeles,CA,U.S.A. https://doi.org/10.2514/6.2001-5226

Taylor, S. J. E. (2019). Distributed simulation: State-of-the-art and potential for operational research. *European Journal of Operational Research*, *273*(1), 1–19. https://doi.org/10.1016/j.ejor.2018.04.032

Tolk, A. (Ed.). (2012). *Engineering principles of combat modeling and distributed simulation*. Wiley Online Library.

Unlu, S. (2015). *Effectiveness of unmanned surface vehicles in anti-submarine warfare with the goal of protecting a high value unit*. Naval Postgraduate School Monterey United States.

Wang, J., Fan, K., Su, Y., Liang, S., & Wang, W. (2008). Air combat effectiveness assessment of military aircraft using a fuzzy AHP and TOPSIS methodology. *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, 655–662. https://doi.org/10.1109/ASC-ICSC.2008.4675442

Weiss, G. (2013). *Multiagent systems: A modern approach to distributed artificial intelligence* (2nd Edition). MIT Press.

Xiao, J., Andelfinger, P., Eckhoff, D., Cai, W., & Knoll, A. (2018). A Survey on Agent-based Simulation using Hardware Accelerators. *ArXiv Preprint ArXiv:1807.01014*.

Zeigler, B. B. (1976). *Theory of Modeling and Simulation*. Wiley Interscience.

Zhao, Z. Q., Hao, J. X., & Li, L. J. (2017). *Study on combat effectiveness of air defense missile weapon system based on queuing theory* (X. Chen, Ed.; p. 1032205). https://doi.org/10.1117/12.2266077