

Lessons Learned in Creating an Autonomous Driver for OneSAF

Dr. Jonathan Stevens, Latika Eifert
Army Research Laboratory (ARL)
Orlando, FL
jonathan.a.stevens@outlook.com
latika.eifert@us.army.mil

Dean Reed, Eugenio Diaz
Institute for Simulation and Training
Orlando, FL
dreed@ist.ucf.edu
ediaz@ist.ucf.edu

Oleg Umanskiy
STILMAN Advanced Strategies
Denver, CO
oleg@stilman-strategies.com

ABSTRACT

The high cost of live training has always been a major challenge for the military. This challenge will only grow as current fiscal uncertainty leads to declining training budgets. Constructive simulations, such as One Semi-Automated Forces (OneSAF), have shown to partially reduce some costs associated with warfighter training. However, further cost reductions in simulation are always sought to ensure that simulation remains an attractive training option for the Commander. The Army Research Laboratory-Human Research and Engineering Directorate, Simulation and Training Technology Center focused on an effort to lessen costs by creating an automatic 'driver' for OneSAF with less need for human intervention. For this initial effort, we describe how the Linguistic Geometry Real-Time Adversarial Intelligence and Decision-making (LG-RAID) lightweight simulation generated and sent to OneSAF tactically valid cooperative entity behaviors for an entire company-size-force of friendly and enemy combatants. We discuss how this was accomplished, for both scenario creation as well as scenario execution. For this initial paper, our results primarily focused on scenario creation, with follow-on studies concentrating on scenario execution. This paper describes key principles developed behind the 'driver' and offers potential areas for future research based upon our lessons learned in this study.

ABOUT THE AUTHORS

Dr. Jonathan Stevens is a Lieutenant Colonel in the United States Army and a Research Program Manager of the Simulation and Training Technology Center (STTC), a subordinate organization of the Research Development and Engineering Command's (RDECOM) Army Research Laboratory (ARL). LTC Stevens has over 22 years of military experience as an Infantry and Acquisition Corps officer.

Latika (Bonnie) Eifert (M.S. in Computer Engineering, UCF 2003). As Science and Technology Manager at the U.S. Army Research Laboratory-Human Research and Engineering Directorate, Simulation and Training Technology Center (ARL-HRED STTC) located in Orlando, Florida, Ms. Eifert manages several projects associated with simulation and training. She is also supporting the Defense Advanced Research Project Agency (DARPA) by managing research program efforts.

Dean Reed (B.S. in Computer Science, UCF 2000) is a Senior Associate for Simulation with the Institute for Simulation and Training (IST) of the University of Central Florida. Mr. Reed is a veteran of the U.S. Army, leads a team of developers at IST and has been using variations of Semi-Automated Forces (SAF) software throughout his 15-year career.

Eugenio Diaz (B.S. in Computer Engineering & Electrical Engineering, University of Puerto Rico 1996) is an Associate for Simulation with the Institute for Simulation and Training (IST) of the University of Central Florida. Mr. Diaz has extensive OneSAF software build experience over his 12-year career.

Oleg Umanskiy (M.S. in Computer Science & Engineering, UCD, 2001) is the Chief Software Architect at STILMAN Advanced Strategies and leads the software development team. Oleg has over 18 years of software design, development and research experience, and has been the architect and the leading developer of all the problem-oriented and generic LG-based systems developed at STILMAN, including generic LG-PACKAGEs licensed to major US and international defense contractors and to British government. He received special recognition from the DARPA RAID management.

Lessons Learned in Creating an Autonomous Driver for OneSAF

Dr. Jonathan Stevens, Latika Eifert
Army Research Laboratory (ARL)
Orlando, FL
jonathan.stevens@us.army.mil
latika.eifert@us.army.mil

Dean Reed, Eugenio Diaz
Institute for Simulation and Training
Orlando, FL
dreed@ist.ucf.edu
ediaz@ist.ucf.edu

Oleg Umanskiy
STILMAN Advanced Strategies
Denver, CO
oleg@stilman-strategies.com

INTRODUCTION

Simulation-Based Training (SBT) continues to expand in the United States Army as well as other militaries throughout the world (Lele, 2013). SBT has been shown to successfully provide transfer of Knowledge, Skills and Abilities (KSAs) from the simulated environment to the environment those KSAs are actually used in (Harrington, 2011) (Blow, 2012) (Seymour, et al., 2002) (Hays, Jacobs, Carolyn, & Eduardo, 1992) (Salas, Rosen, Held, & Weissmuller, 2009). However, in an era of fiscal constraint, the cost of simulation must be reduced (Insinna, 2013) (Insinna, 2014), particularly for the U.S. Army (Thompson, 2013).

There are three classes of simulation used for training: live, virtual and constructive (Hodson & Hill, 2013). In the past decade, a fourth class has been added: gaming (Roman & Brown, 2008). Live simulations comprise real people operating real systems while virtual simulations encompass real people operating simulated systems in simulated environments. Constructive simulations involve simulated people operating simulated systems. The term gaming refers to the employment of interactive, computer-based applications used for training purposes (Bergeron, 2006), generally characterized by their low overhead and cost. When the four classes overlap each other in training (i.e. employment of more than one class in a training exercise), this is referred to as "blended training".

One of the major advantages of employing virtual and/or constructive simulation in training is its cost advantages (Orlansky, et al., 1994), especially when compared to live training. Constructive simulation, in particular, has been found to be a significant cost-saving option (Riecken, et al., 2013), particularly considering today's constructive simulations can employ thousands of entities simultaneously in the simulation.

BACKGROUND

The Army's One Semi-Automated Forces (OneSAF) simulation is an entity-level, composable, constructive simulation used to portray the Contemporary Operating Environment (COE) (Parsons, Surdu, & Jordan, 2005). OneSAF provides accurate units, behaviors, effects and the synthetic environment that enable users, as well as other simulations, to execute constructive SBT. The simulation is designed for the brigade and below level echelon, supporting both combat and non-combat operations. OneSAF is currently used by the Army's Advanced Concepts and Requirements (ACR), Research, Development and Acquisition (RDA) and Training, Exercises, Military Operations (TEMO) communities.

The Linguistic Geometry Real-time Adversarial Intelligence and Decision-making (LG-RAID) light-weight simulation leverages novel game theory to generate intelligent, predictive and tactically-correct courses of action (COAs) for constructive exercise participants (Stilman & Yakhnis, 2003). LG-RAID entities include friendly, enemy and civilian forces. Similar to OneSAF, the LG-RAID targeted training echelon is brigade and below and provides the user with accurate units, behaviors and effects in the simulated environment. Recent U.S. Army efforts

are focused on modifying this capability to a cloud-based service, capable of rapid and persistent (i.e. 24/7) COA analysis for the end-user.

While LG-RAID is capable of operating as a standalone simulation, use cases exist when high-fidelity, entity-level simulations, such as OneSAF, are required to meet the user's training requirements. However, OneSAF does not possess the robust predictive course of action (COA) capability that LG-RAID provides. To realize the benefits of both simulations, we interoperated the two products, using LG-RAID to serve as an automatic 'driver' for OneSAF. In doing so, we reduced the need for human role players, and thereby reduced the cost to conduct the training exercise.

In this paper, we describe how the LG-RAID capability to automatically compute, and subsequently send, intelligent guidance to OneSAF entities was accomplished, for both scenario creation as well as scenario execution. For this initial paper, our emphasis will focus on scenario creation. Follow-on studies will concentrate on scenario execution, which proved to be beyond the scope of this initial effort. Additionally, we will describe key principles developed behind the 'driver' and offer potential areas for future research based upon our lessons learned in this study.

METHOD

Purpose

The purpose of this effort was to develop and interoperate LG-RAID capabilities to serve as an intelligent light-weight 'driver' for the OneSAF simulation system. Since OneSAF is a "Semi" Automated Forces simulation system, it requires human operators (often referred to as "pucksters") to set up and control entities within the simulation. Each operator can control a group of entities (e.g. a squad or platoon) by employing the semi-automated behaviors of the units, however a certain degree of human management is typically required. For a typical exercise, this can result in a significant manning requirement to control both the friendly and enemy units in the simulation and constitutes a significant overhead cost.

In contrast, LG-RAID provides a simulation capability with fully automated tactics and behavior generation (with an option for human-in-the-loop participation). Leveraging LG-RAID's capability to automatically generate intelligent, tactically valid courses of action for all entities allows for the reduction in manpower needed to execute a OneSAF scenario, by reducing the need for human role players. In addition, LG-RAID's feature set includes the generation of most-likely/most-dangerous initial placement of enemy forces. This can further simplify deployment of a OneSAF exercise by "fast-tracking" the creation of a OneSAF scenario with automated initial conditions. Thus, the following technical goals were identified for this effort:

- Reduce the OneSAF personnel requirement for scenario creation, by automatically placing enemy forces in the most dangerous/likely positions at scenario initialization, and thus provide a challenging scenario for the BLUFOR commander.
- Reduce the OneSAF personnel requirement for scenario execution by automatically, without human intervention, controlling units within OneSAF through the generation and transmission of intelligent and tactically correct commands for each unit.

Interoperability Methodology

While interoperability can be achieved through the use of either the High Level Architecture (HLA) or Distributed Interactive Simulation (DIS) protocols, for this experiment we relied upon OneSAF's User Data Gateway (UDG). The UDG provides access to internal OneSAF data and better supports interoperability between OneSAF and external applications. The UDG's expanded functionality enables commands to be sent from LG-RAID to OneSAF (and vice versa), thus enabling the continuous, real-time updates required to conduct distributed simulation. The UDG allowed LG-RAID to publish, retrieve, and update both OneSAF and itself.

To avoid correlation issues, both simulations utilized the same OneSAF Terrain Format (OTF) data. While LG-RAID employs a different internal terrain data model, ingesting OTF terrain data into LG-RAID ensured close terrain correlation and reduced potential fair fight discrepancies, resulting in more accurate results. Specifically, the Horn of Africa (HOA) OTF database was utilized, consisting of a small urban environment with Ultra High

Resolution Buildings (UHRB) models. This allowed for a wider spectrum of small unit operations – including entering into and firing from within structures. Furthermore, both simulations employed the same models for entities, units and behaviors, thus further reducing typical interoperability "noise" from the experiment. Specifically, this experiment's use cases employed the following entities and units:

- Multiple BLUFOR platoons. Each platoon is further decomposed into:
 - Three rifle squads, each consisting of a squad leader with two fire teams composed of:
 - Team Leader
 - SAW gunner
 - Grenadier
 - Basic rifleman
 - One weapons squad, each consisting of:
 - Squad leader
 - M240 gunner
 - M240 assistant
 - Javelin gunner
 - Javelin assistant gunner
- Nine OPFOR units, each consisting of:
 - Commander
 - Two riflemen

By using the same terrain database, similar entities/unit models and the OneSAF UDG, efficient bi-directional information exchange between both simulations was achieved. Such interoperability not only improves the ease-of-use for scenario setup, but also allows the user to determine which simulation "drives" which. Scenario creation can occur in either simulation and when completed, can be ported from one simulation to the other with the appropriate initial conditions. In the two experiments outlined below, we chose one simulation to create and 'drive' the scenario for both use case #1 and use case #2. We subsequently analyzed which simulation allowed for a more efficient creation of the two different use cases. Follow-on papers will analyze which simulation provided a more efficient and effective apparatus for scenario execution.

Interoperability

The initial exchange of scenario data (prior to simulation execution), using the LG-RAID interface through the UDG, allows both systems to create friendly or hostile entities and share the information with the other system so as to produce identical starting conditions. LG-RAID employs a translation layer, using a set of translation tables, to provide mapping between LG-RAID entities, units, and weapons and OneSAF entities, weapons, and compositions. Bi-directional scenario setup information exchange additionally allows the leveraging of LG-RAID's capability to generate enemy forces in the most dangerous/likely initial positions, and thus provide a challenging scenario for the BLUFOR commander. New units generated by LG-RAID are translated into OneSAF unit compositions and distributed into OneSAF via the UDG. Initial unit positions specified by the human operator in either or both systems can be expanded by automatically adding additional enemy forces in key locations.

In contrast, when creating a OneSAF scenario in the traditional mode, human operators use the OneSAF Management and Control Tool (MCT) in order to manually create and emplace entities, control measures and assign tasks based on the overall mission for the scenario. This placement and tasking is performed based on the overarching mission objective provided by the Commander. Figure 1 graphically compares the scenario creation process using OneSAF natively (via the MCT) and using the LG-RAID interface (via the UDG).

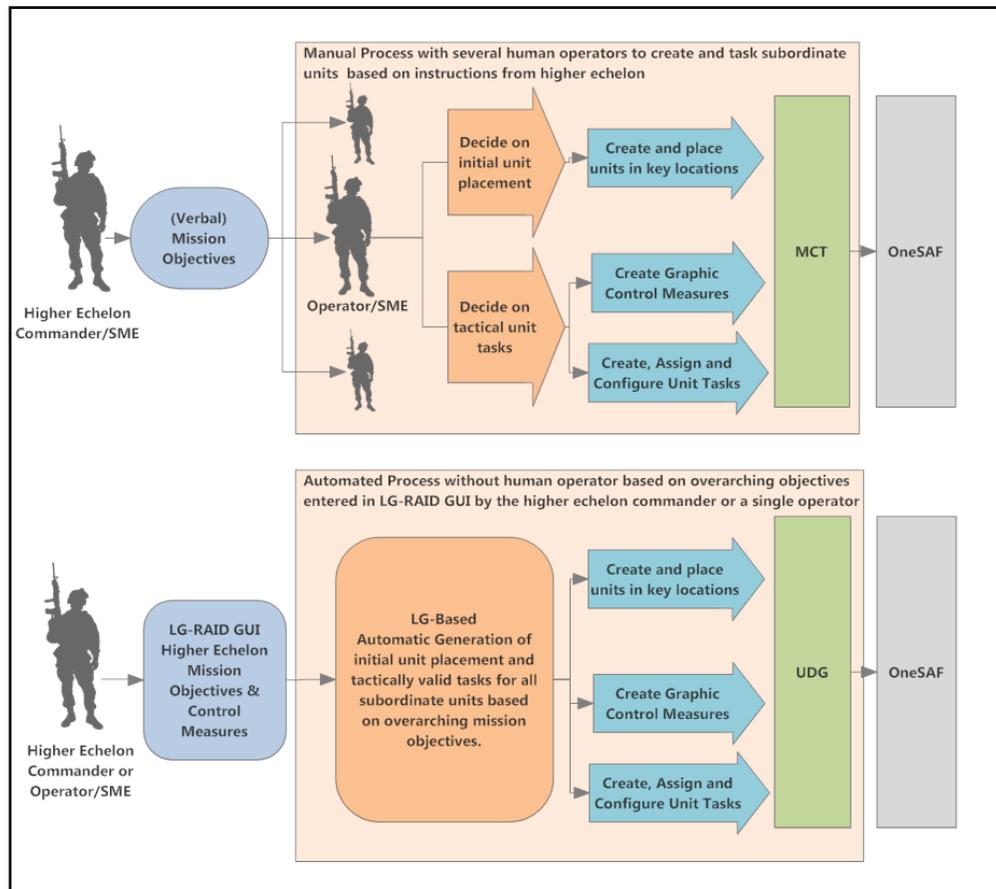


Figure 1. Scenario Creation using OneSAF Natively and LG-RAID

When LG-RAID "drives" the distributed simulation event, a number of autonomous events occur. Once the initial data exchange has occurred, LG-RAID generates cooperative entity behaviors for an entire company-size-force of friendly and/or enemy combatants. The user can provide various levels of input to guide the COA generation, such as graphic control measures for subordinate units. LG-RAID will then generate the coordinated plan for all the entities within the subordinate units. For example, the user may choose a generic tactical task (such as "harass and interdict") and then allow LG-RAID to generate tactically relevant, coordinated actions for the entire enemy and friendly force. LG-RAID then dynamically and intelligently responds to both the BLUFOR and OPFOR actions – similar to how human operators would control OneSAF entities and units. To achieve such level of coordination, LG-RAID only sends small portions of the plans, then monitors the execution of these plans, and sends additional tasks as needed to keep the entire force synchronized.

After the initial tasks have been transmitted and scenario execution commences within OneSAF, LG-RAID polls the state of the simulation at a user-specified frequency, which allows for both simulations to stay synchronized. OneSAF is considered to be the 'master' simulation – all movement, weapon engagements, and other events are adjudicated within OneSAF and the results used by LG-RAID to maintain a consistent current operating picture. LG-RAID employs its internal simulation model only to estimate the outcome of events to continuously calculate the COAs. Based on the updated simulation state, LG-RAID rapidly re-evaluates COAs and makes a determination whether the tasking for units needs to be updated. If needed, the new set of OneSAF tasks is sent through the UDG to replace the previous orders. LG-RAID computational speed allows for such adjustments to be done in a timely manner.

The key characteristic of this approach is that LG-RAID is able to generate updated COAs in response to the changing simulation state without additional human input. Based on the original 'high-level' missions specified at

the beginning of the exercise, such as ‘harass & interdict’, specific coordinated and tactically valid actions are generated that dynamically adjust to the changes in the enemy’s COAs and behaviors. Figure 2 graphically compares the scenario execution process using OneSAF natively (via the MCT) and using the LG-RAID interface (via the UDG).

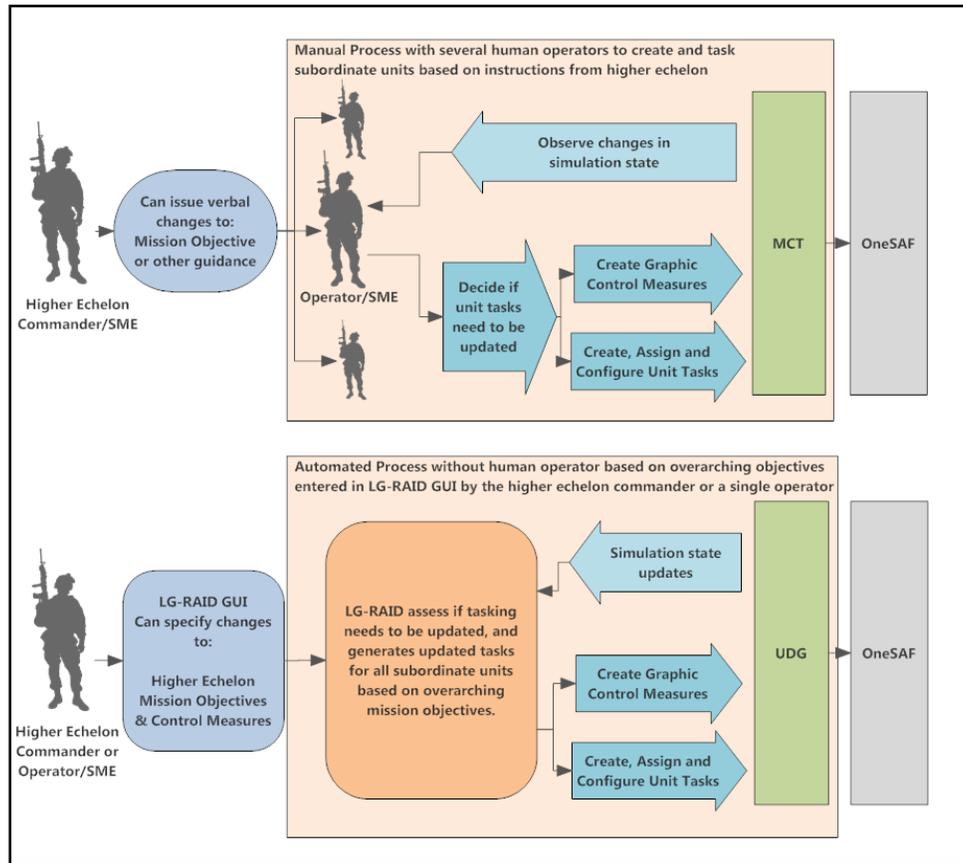


Figure 2. Scenario Execution using OneSAF Natively and LG-RAID

Baseline Configuration and Architecture

This experiment was conducted using one laptop, virtualized to replicate distributed simulation between two machines connected via LAN; with one virtualized machine running LG-RAID and the other running OneSAF version 6.1. Laptop specifications are listed in Table 1.

Table 1. Laptop Specifications

Brand:	Fujitsu Lifebook
CPU:	2.8 GHz i7-2640 M
RAM:	8GB
Disk:	500GB
OS:	Windows 7 x64

Use Case #1 Description

In scenario one (Figure 3 and 4), one BLUFOR platoon (see composition above) moved tactically through the town of Las Khoreh to its designated objective. During movement, the BLUFOR platoon made contact with three enemy

elements (total of 27 personnel) and subsequently reacted to contact. Scenario #1 was executed first with LG-RAID serving as both the scenario creation platform as well as the 'driver' of the distributed simulation. We then created and executed the same scenario using core OneSAF behaviors and employing OneSAF as the simulation driver.

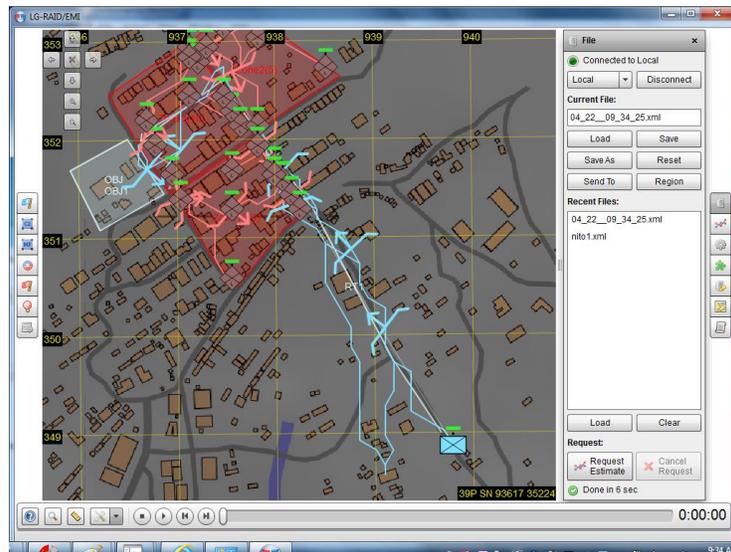


Figure 3. Scenario #1 (LG-RAID)

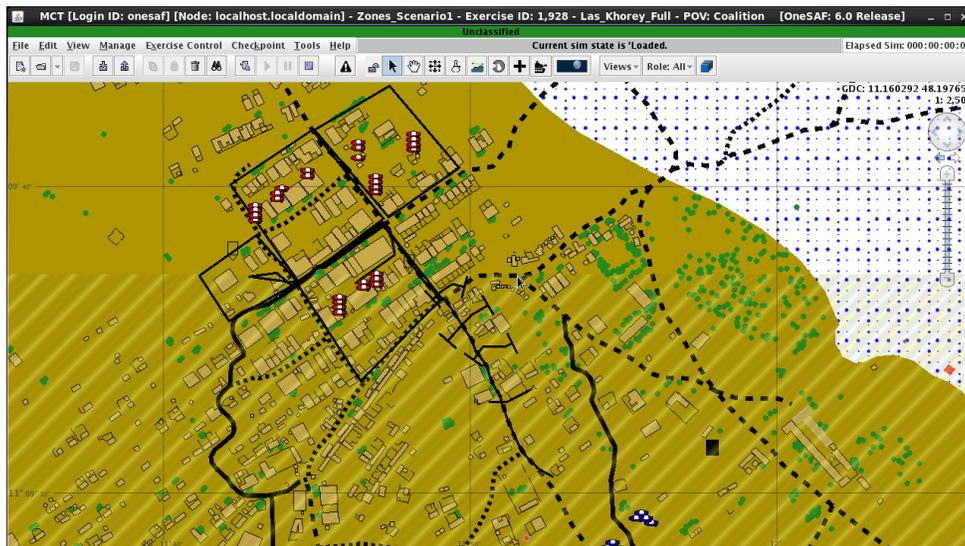


Figure 4. Scenario #1 (OneSAF)

Use Case #2 Description

We took an alternative approach to the scenario development in use case #2 and chose a tactical situation that exploited some of the higher functional capabilities of OneSAF. In this scenario (Figures 5 & 6), two BLUFOR fire teams conducted a presence patrol at a road intersection in Las Khoreh. The BLUFOR were then engaged by enemy forces intermixed with the local populace exactly 20 minutes after their arrival at the intersection. The OPFOR

elements consisted of a nine-man terrorist cell dismounting from a civilian truck that approached the BLUFOR from a concealed route.

Similar to before, Scenario #2 was executed first with LG-RAID serving as both the scenario creation platform as well as the 'driver' of the distributed simulation. We then created and executed the same scenario using OneSAF as both the scenario creator and the simulation driver.



Figure 5. Scenario #2 (LG-RAID)

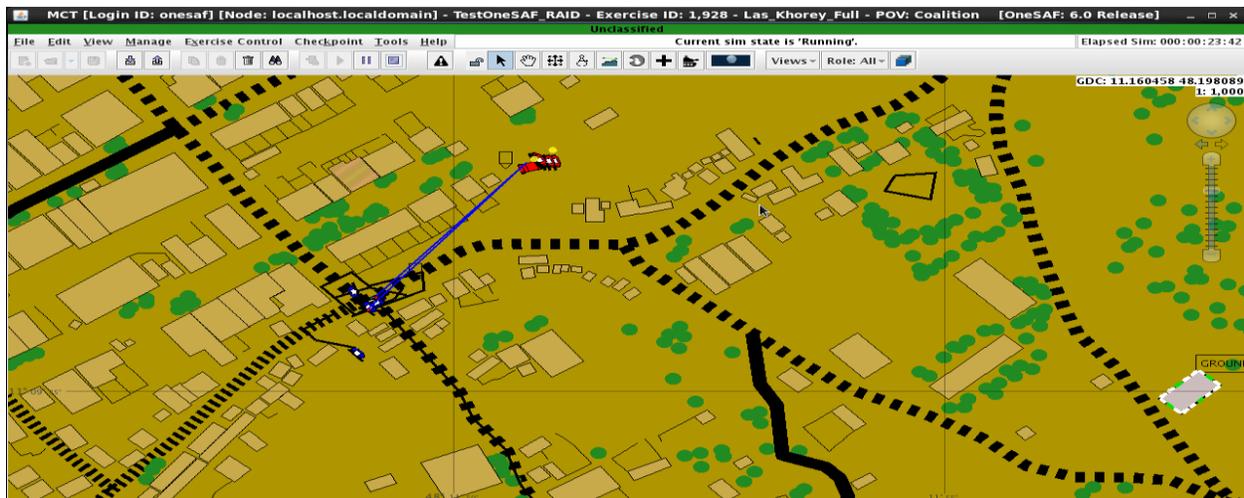


Figure 6. Scenario #2 (OneSAF)

RESULTS

Use Case #1 Results

Two experienced OneSAF developers created and executed scenario #1 using LG-RAID as the distributed simulation 'driver' and then OneSAF as the 'driver'. Scenario creation results are depicted in Table 2. Per Table 2,

scenario creation was far quicker (for both developers) using LG-RAID as the simulation driver. Manually re-expressing complex LG-RAID automatic behaviors, such as the “OPFOR Occupy”, with OneSAF atomic behaviors was very time intensive and prone to error. For this use case, a very complex execution matrix was created in OneSAF whereas LG-RAID's execution matrix was simple to create. The OneSAF execution matrix required the creation of 3 subordinate execution matrices (with one matrix extending 14 phases) and required 24 start-to-finish runs to ensure the scenario executed properly, whereas LG-RAID only required one pre-execution run. Scenario execution performance was approximately equal between both simulations and will be explored further in a follow-on study.

Table 2. Use Case #1 Results

User	LG-RAID Scenario Creation Time	OneSAF Scenario Creation Time
OneSAF Developer # 1	25 Minutes	17 Hours, 20 Minutes
OneSAF Developer # 2	40 Minutes	7 Hours, 35 Minutes

Additionally, for Use Case #1, we examined the ease-of-use of both simulations during scenario creation by evaluating the level of effort required to input various graphic control measures. Prior to scenario execution, in accordance with FM 1-02 (Operational Terms and Graphics), our two experienced OneSAF developers emplaced routes, Attack by Fire (ABF) positions and other graphic control measures required to execute the scenario in two conditions. Similar to before, the first condition again had LG-RAID as the distributed simulation 'driver', with the second condition using OneSAF as the 'driver'. Table 3 depicts our results, with LG-RAID representing a more efficient scenario creator than OneSAF.

Table 3. Use Case #1 Results

User	LG-RAID Graphic Control Measure Creation Time	OneSAF Graphic Control Measure Creation Time
OneSAF Developer # 1	0 Minutes	44 Minutes
OneSAF Developer # 2	0 Minutes	36 Minutes

Use Case #2 Results

The same two experienced OneSAF developers created and executed scenario #2 using LG-RAID as the distributed simulation 'driver', followed by OneSAF serving as the exercise's driving simulation. Scenario creation results are depicted in Table 4 and show that scenario creation time was approximately equal between both simulations. Scenario execution performance was again approximately equal between both simulations and will be explored further in a follow-on study. However, it was obvious for this use case that OneSAF was able to portray scenario execution results and events at a much finer level of granularity than LG-RAID.

Table 4. Use Case #2 Results

User	LG-RAID Scenario Creation Time	OneSAF Scenario Creation Time
OneSAF Developer # 1	30 Minutes	75 Minutes
OneSAF Developer # 2	40 Minutes	60 Minutes

DISCUSSION

Conclusions

While OneSAF affords the user a finer level of granularity than LG-RAID, the additional scenario creation time required may not be worth that additional level of detail and control. Scenario #1, in particular, highlighted the advantages of employing LG-RAID as the exercise driver, reducing scenario creation time drastically when

compared to using OneSAF in that role. This, in turn, can potentially result in significant cost savings by reducing the need for human operators during scenario creation. The capability to use LG-RAID as a tool to rapidly set-up the initial conditions of a complicated scenario, and then save it as a native OneSAF scenario, is a noteworthy finding that merits further research. Scenario execution will also be examined in a later study.

To summarize, based upon this experiment's results, we believe we have demonstrated potential significant scenario construction time savings when employing LG-RAID as the distributed simulation 'driver'. No appreciable performance differences were discovered during simulation exercise between both treatments (i.e. LG-RAID as the simulation driver and OneSAF as the simulation driver); however this will be examined more thoroughly in a follow-on study.

This research effort has the potential to change the manner in which the Army executes its entity-level, constructive simulation training exercises. By employing a novel technology and approach, we demonstrated that LG-RAID has the potential to achieve significant time and cost savings, while still retaining the strengths of OneSAF. Scenario creation, in particular, represented the area that benefited the most from LG-RAID's capabilities.

Limitations

Since this was a research effort, only a limited set of OneSAF and LG-RAID functionality was reviewed. Additionally, this effort does not seek to replace, but rather augment, the functionality of OneSAF by using a game-based lightweight simulation to potentially reduce labor intensive tasks when creating scenarios.

Recommendations for Future Research

Our first recommendation for future research is to examine the effect on scenario execution performance when employing LG-RAID as the simulation 'driver'. While significant time savings were achieved during scenario creation, further analysis must be conducted in order to ascertain LG-RAID's effect on scenario execution. This recommendation requires significant planning and resources, which was outside of the scope of this current effort. Additionally, the exploration of a hybrid approach, whereby the 'driver' of the exercise switches between simulations, in real-time, so as to take advantage of each simulations' inherent strengths would be a noteworthy endeavor. Network load balancing and exercise scaling experimentation present additional topics for potential future research efforts. Finally, the integration of LG-RAID into other existing simulations should be explored so as to determine if similar cost and performance improvements can be discovered.

REFERENCES

- Bergeron, B. (2006). *Developing Serious Games (Game Development Series)*. Charles River Media.
- Blow, C. (2012). *Flight School in the Virtual Environment*. Fort Leavenworth, Kansas: United States Army Command and General Staff College.
- Harrington, M. (2011). Empirical Evidence of Priming, Transfer, Reinforcement, and Learning in the Real and Virtual Trillium Trails. *Learning Technologies IEEE Transactions on*, 175-186.
- Hays, R., Jacobs, J., Carolyn, P., & Eduardo, S. (1992). Flight Simulator Training Effectiveness: A Meta-Analysis. *MILITARY PSYCHOLOGY*, 63-74.
- Hodson, D. D., & Hill, R. R. (2013). The Art and Science of Live, Virtual and Constructive Simulation for Test and Analysis. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*.
- Insinna, V. (2013, July). Simulation, Gaming Sector Plagued by Fiscal Challenges. *National Defense*, pp. 22-23.
- Insinna, V. (2014, February). Defense Simulation Firms Turn to Commercial Sector for Inspiration. *National Defense*.
- Lele, A. (2013). Virtual Reality and its Military Utility. *Journal of Ambient Intelligence and Humanized Computing*, 17-26.
- Orlansky, J., Dahlman, C., Hammon, C., Metzko, J., Taylor, H., & Youngblut, C. (1994). *The Value of Simulation for Training*. Alexandria: Institute for Defense Analyses.
- Parsons, D., Surdu, J., & Jordan, B. (2005). OneSAF: A Next Generation Simulation Modeling the Contemporary Operating Environment. *Euro-Simulation Interoperability Workshop*.

- Riecken, M., Powers, J., J. C., Numrich, S. K., Picucci, P. M., & Kierzewski, M. (2013). The Value of Simulation in Army Training. *The Interservice/Industry Training, Simulation & Education Conference (IITSEC)*. National Training Systems Association.
- Roman, P., & Brown, D. (2008). Games – Just How Serious Are They? *Interservice/Industry Training, Simulation, and Education Conference (IITSEC)* (pp. Vol. 2008, No. 1). NTSA.
- Salas, E., Rosen, M., Held, J., & Weissmuller, J. (2009). Performance Measurement in Simulation-Based Training: A Review and Best Practices. *Simulation & Gaming*, 328-376.
- Seymour, N., Gallagher, A., Roman, S., O'Brien, M., Bansal, V., Anderson, D., & Satava, R. (2002). Virtual Reality Training Improves Operating Room Performance. *Annals of Surgery*, 458-464.
- Stilman, B., & Yakhnis, V. (2003). Linguistic geometry: new technology for decision support. *AeroSense 2003*. International Society for Optics and Photonics.
- Thompson, M. (2013, November 4). Reshaping the Army: A Force Built to Fight the Cold War is now Battling Changes to its Size, Shape and Mission. *Time*, pp. 34-40.