A New Approach to Building Agile Simulations

Charles G. Sanders
Edge360 LLC
Suffolk, VA
csanders@edge360.com

ABSTRACT

Technology and threat change is outpacing Defense Acquisition and modeling and simulation (M&S) development processes. The current M&S architectures do not support agility. DoD struggles with interoperability between simulations and reuse of models and data. Models and data generated in new capability design and development are often not available for testing, analysis or training systems development. M&S Community stovepipes inhibit data and model sharing and reuse, partially because M&S is funded and managed separately by each community. An opportunity has emerged with the embrace of model-based systems engineering (IAW DoD Digital Engineering Strategy), where digital artifacts from system design and development should be available for use by other applications (such as testing, analysis, and training) across the system acquisition lifecycle. However, there is no common modeling framework or infrastructure to capture the data and models to maintain more coherent representation of the system in testing, analysis, experimentation, and training applications. Also, DoD lacks a process or infrastructure to rapidly develop and deliver new M&S capabilities based on emerging technologies and threats. This paper explores current DoD MBSE implementation efforts and existing modeling frameworks that could provide a more cost-effective way to model the performance and behaviors of new systems in an operational environment simulation and provide an infrastructure for agile composable simulations to facilitate coherent, traceable modeling across a system lifecycle.

ABOUT THE AUTHOR

Dr. Charles Sanders is a Principal Modeling and Simulation Analyst for the Defense Modeling and Simulation Coordination Office and Army Modeling and Simulation Office. With over 30 years of experience in modeling and simulation, organization development, management and leadership, strategic policy and planning, and training and learning concepts; to include corporate staff support at all levels, he has provided advice to Office of Secretary of Defense, Joint Chiefs of Staff, U.S. Navy Staff, Joint Forces Command, and other executive level government and industry. Dr. Sanders led studies of emerging technologies for the M&S community, including cloud-based simulation, modeling frameworks and development of a persistent distributed simulation test bed for FVEY partner nations. He previously provided strategies and plans support to Joint Staff J7 and ODUSD-P&R, including initiation and development of the Department of Defense Training Transformation Program, and strategic support of the DoD Advanced Distributed Learning Initiative. Dr. Sanders also served as the requirements and resource sponsor for Joint Simulation System (JSIMS) Maritime.

A New Approach to Building Agile Simulations

Charles G. Sanders
Edge360 LLC
Suffolk, VA
csanders@edge360.com

INTRODUCTION

Simulations in the Department of Defense (DoD) is resourced and managed by separate communities. Most of the existing simulations were developed in those stovepipes as standalone capabilities tailored to each community requirements (McGroder & Lashlee, 2011). Given the distinct requirements of each community, and the state of technology when most of the existing modeling and simulation (M&S) capabilities were developed, integration for cross community cooperation is technically and fiscally challenging. This means the same models and data required to support each community simulation is created or collected separately, with very limited ability to share or leverage the investments of others. This reality is likely to persist, because the DoD lacks a coherent process and infrastructure to rapidly develop and deliver performance or behavior modeling of new systems and deliver new simulation capabilities based on emerging technologies.

According to Culton et al (2016) federated architectures are outdated: vulnerable and inefficient. The current approach for simulation supported training is to establish a net-centric interoperable capability through the federation of multiple and diverse simulations. Culton et al (2016) further explained that today's tools and technical infrastructure:

- Are predominantly monolithic (all or nothing)—limited or no ability to compose or tailor unique training environments to meet specific trainer/trainee objectives.
- Are built to differing standards that require significant time and specialized contract labor to integrate into useful federations that support exercises or training events.
- Are expensive to operate and sustain.
- Lack the agility to rapidly reflect changing operational environment and emergent threats.
- Are technically complicated, limiting the ability to make effective change to the synthetic representation of the operational environment.
- Contain redundant capabilities.
- Provide support for discrete events versus continuous on-demand (24/7) accessibility.
- Are increasingly under pressure to meet growing infrastructure consolidation and cybersecurity compliance mandates.

According to Kewley & Sapol (2014), current practices for federated simulation make analysis of systems performance very difficult. Distributed Interactive Simulation and High-Level Architecture are integration technologies that only share minimal state data. Simulation initialization, simulation state, time, terrain, and models must be coordinated across federates. Coordinated simulation initialization is complex. Input data is hidden in simulation specific data files. Given any complex scenario with more than two or three federates, analysis is painstaking and fraught with errors. Verification and validation of the federation is also an immense challenge.

Modifying or updating legacy M&S tools is not sufficient to meeting emerging requirements, as they are inherently man-power intensive, requires coding skills, and costly to upgrade and sustain (Culton et al, 2016). Current design and reliance on proprietary solutions sustains stovepipes between the different M&S communities and does not support natural interoperability or sharing of M&S resources. Therefore, a new approach is required to develop the required capabilities more rapidly and with much less post-development integration reengineering.

The lack of agility is particularly significant, given the complex, unpredictable, and dynamic nature of the forecasted operational environment descripted by Billings et al (2016), where a more agile and rapid M&S development environment is required. The DoD M&S Enterprise requires new M&S capabilities to meet future warfighting requirements and accurately represent complex and dynamic environments and threats. Some of the new required capabilities are due to changes in information technologies, such as migration to cloud computing, parallel processing, and M&S as a Service. Also, the M&S Enterprise is challenged by the reality that M&S technology is changing and opportunities are emerging faster than the current paradigm can respond. Therefore, a new integrated approach to providing simulations is essential.

M&S AS A SERVICE

The NATO M&S Group (Hannay & van den Berg, 2014; Siegfried et al, 2014) noted that recent technical development in cloud computing technology and service-oriented architecture (SOA) offers opportunities to utilize M&S capabilities better in order to satisfy NATO critical needs. M&S as a Service (MSaaS) is based on the SOA approach and potentially offers a means of delivering M&S applications, capabilities, and associated data on demand (Siegfried & van den Berg, 2015). MSaaS is intended to promote discovery, reusability, and composability of M&S services. One key aspect of MSaaS is use of modularized simulations and tools that can be easily tailored for each use case, rather than rely on a few monolythic simulations that provide all the required simulation functionality or representation. This approach enables executing each distributed simulation use case with less required resources and data streams. However, a key requirement for the service-oriented simulation environment is composability (Tolk & Mittal, 2014).

COMPOSABILITY

Kasputis and Ng (2000) argued that model reuse is dependent on composability; the models must be designed to work together. They further argued for a 'robust, theoretically grounded framework for design' (pg. 1). That framework could be what may emerge from the DoD transition to digital engineering. RAND defined composability as the capability to select and assemble components in various combinations to meaningfully satisfy specific user requirements (RAND, 2003). Bartholet et al (2004) defines composability as the capability to select and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements.

Bartholet et al (2004) argued that DIS, HLA, and ALSP have two significant shortfalls with respect to composability: the inability to guarantee consistency between simulations in the federation, and the inability to provide for component reuse in other federations without significant source code modifications. In other words, while these tools can provide the means for exchanging and maintaining entity state, resolving interactions, managing time, and managing data distribution, they provide only minimal facilities for ensuring the federation is a meaningful simulation. This lack of support often results in simulations that fail to meet requirements because of a lack of consistency caused by fundamental differences in underlying models. Bartholet et al (2004) concluded that "if components are to be composable, they have to be designed for it." One way to enable composable simulation is to start with a common modeling framework.

OPPORTUNITY – NEW MODELING FRAMEWORKS

New modeling frameworks have emerged that offer an opportunity for sufficient composability and to better leverage new technologies and tools, and create inherently interoperable capabilities faster through cooperative and coordinated R&D efforts (Blair, Sholes & Reed, 2016; Clive et al, 2015). These modeling frameworks enable rapid scenario development with the ability to quickly build models of new systems and then play them in the scenario. Open source software development promises to enable crowd sourcing, proven to be much more effective for innovations. Some of the available modeling frameworks are described below.

AFSIM

AFSIM is a government-approved software simulation framework for use in constructing engagement and mission-level analytic simulations for the Operations Analysis community (Clive et al, 2015; AFRL, 2016). The primary use of AFSIM applications is the assessment of new and advanced system concepts, and the determination of concepts of employment for those systems.

The framework provides the ability to model the capabilities of the participants and to control the interaction of the participants as they move through space and time. The resulting simulations can be:

- Constructive/non-interactive (the user invokes the simulation which then runs without further interaction), or interactive (the user or other simulation controls some aspects of the simulation).
- Non-real-time (faster or slower depending on the fidelity of the platform component models), or real-time (constrained by some multiple of a real-time clock).
- Event-stepped (simulations proceed according to processing of relevant events) or time-stepped (simulations proceed according to events occurring in succeeding time steps)

AFSIM is designed to be generally usable "as is," so that not only are top-level modeling concepts defined by the framework, but many concrete implementations are also provided. Of note, many models are offered at different levels of detail or fidelity to enable reducing the demand on computing power where detail is not required. In addition to all the models made available by the community of developers and users, examples of standard models delivered with the framework include the following:

- Movement models
- Sensor systems
- Weapon systems and weapon effects
- Communication systems
- Information processing systems (trackers, etc.)
- Decision making systems (command and control, missile guidance, etc.)
- Antenna pattern models
- Atmospheric attenuation models
- Signal propagation models
- Clutter models
- Electronic warfare effect models

AFSIM is also extensible, providing for ease of incorporating models of the above types, as well as completely new capabilities. Because AFSIM is a framework, simulation details and services are provided, freeing the developer to focus specifically on adding desired new functionality through implementation of the framework's abstract interfaces. Additionally, AFSIM's Component-Based Architecture (CBA) provides the ability to cleanly and generically incorporate nearly any new modeling capability into the framework. Once incorporated, the models are then as much a part of the framework as any of the standard models. AFSIM also enables selecting models and data at various levels of fidelity to optimize the balance of performance.

EASE DMF

The Executable Architecture for Systems Engineering (EASE) Distributed Modeling Framework (DMF) is intended to lower the human and material costs of leveraging M&S (Snively, Leslie & Gaughan, 2013), and provides a user interface for the execution of simulation experiments in a set of cloud-based computing resources (Kewley & Sapol, 2014). EASE DMF is based on discrete event simulation framework (DEVS) (Zeigler, 2000), which is designed to simulate component-based systems, and is a good match for service-oriented simulation (Sarjoughian et al, 2008). The EASE DMF provides a different way to build federations. In order to compose models, the developer must first break them apart from their containing simulations– assembling a set of models required by the systems of interest. Each component is a Discrete Event Systems Specification (DEVS) atomic model of a system function. Computations are extracted from existing simulation models as pure state transition functions with no side effects. The framework provides the ability to manage initialization properties, state, scheduling, and parallel execution. Each component exposes its initialization properties, state, and event message interfaces via Extensible Markup Language (XML) in order to enable code generation and to support identification of parameters that can be linked to system architecture properties or to performance metrics. EASE DMF guides the development of models, the integration via event messages, and the execution in support of an experimental design. A reference implementation of EASE-DMF uses the actor model of computation within the Akka actor framework (https://akka.io/). This framework has a natural alignment with the message-based nature of DEVS, and it has built in support for asynchronous computation, distribution of models across networks, and parallel execution. A model scenario integrates existing target acquisition, communication, and terrain models in a simulation of soldier situation awareness. This framework has not been tested for large scale simulation. So, experimentation would be required to validate scalability for operational or strategic level applications.

WarpIV Kernel

The WarpIV Simulation Kernel is a modeling framework that provides high performance parallel discrete-event and real-time simulation capability (www.warpiv.com). The WarpIV Kernel promotes an open-source component-based plug and play modeling paradigm to integrate reusable models.

The WarpIV Kernel hosts discrete-event simulations over parallel and distributed cluster computing environments. WarpIV provides a next-generation object oriented computational framework that combines state-of-the-art distributed event processing with powerful modeling constructs and support utilities. This delivers scalable performance across different computer and network architectures, while simplifying the effort of constructing models for simulation developers. WarpIV Kernel also supports heterogeneous networked applications and executes over Windows (XP and Vista) and Linux/Unix (PC, Mac, SGI, Sun, HP, etc.) platforms.

At the core of WarpIV Kernel is its portable high-speed communications infrastructure that integrates both shared memory with standard network protocols to facilitate high bandwidth and low latency message passing services. Architectures ranging from single/multi-CPU personal computers, to large-scale parallel compute clusters are seamlessly integrated to offer maximum flexibility and portability. WarpIV also supports the exploration of multiple behavior timelines within a single simulation execution. As behavior branching occurs, all common computations between parallel timelines are shared, resulting in an extremely efficient framework for exploring large numbers of behavior permutations without resorting to brute-force Monte Carlo methods.

The only possible drawback of WarpIV is that the expected performance is predicated on all the models being natively developed in the framework and run locally, which means that a user would need to build all the required models in WarpIV, and performance would be suboptimal for distributed simulation.

OSM

Orchestrated Simulation through Modeling (OSM) framework (Winfrey et al, 2014), invented and patented by NSWCDD, provides a framework and interface for modeling system behavior and interactions by "plugging in" models from government, industry or academia. "Plug-in" is a term for a software extension that adds capability or new features. Adding models via plug-ins allows software customization for each project without making changes to the framework. With the OSM framework, an evolutionary system of systems can be intelligently created by a community. Each community member only needs to fully understand the pieces they personally develop. The performance and scalability of OSM framework is discussed in an evolutionary system of systems domain pointing to its efficiency and usability. In addition, OSM enables a mission-level, fast-running, qualitative assessment of the systems' interactions. This assessment permits more focused follow-on analytical deep dives with specialized tools.

OSM uses behavioral models, (i.e., the plug-ins), that represent the systems as independent entities and that follow defined state diagrams where the system moves from state to state by stimulus from other systems or the environment. This approach allows for emergent behavior to be observed as systems interact with each other and the environment throughout a scenario. The behavior is not predetermined or scripted. Each system's behavior can be unique for each run since it is dependent on its interaction with other entities within the model.

OneSAF

While most may not think of One Semi-Automated Forces (OneSAF) as a modeling framework in the same context as those previously described, it is a computer-generated forces simulation that provides entity-level models and behaviors that are both semi-automated and fully automated applications designed to satisfy a broad set of simulation requirements (https://www.peostri.army.mil/onesaf). OneSAF is designed to provide the latest physics-based modeling and data collection, and reporting capabilities. OneSAF models real-world representations of platforms, soldiers, equipment, logistical supplies, communications systems and networks, emerging threats, and aviation assets to achieve the level of fidelity required for a particular application or scenario. The Product Line Approach within the OneSAF (Bartholet et al, 2004) simulation framework offers a more composable framework. OneSAF provides an integrated system for planning, generating, and managing a simulation composed of components built to support the framework. OneSAF is an HLA compliant system, so it has all the interoperability tools that come with the HLA specification. Additionally, OneSAF provides a host of tools for composing a simulation from a set of behaviors, entities, units, and environmental models, all architected to conform to the OneSAF framework. The advantage of the OneSAF framework is that the set of models from which to construct a simulation have all been engineered to work within the framework. Therefore, as long as developers stay within the framework, they should have the ability to modify these models to meet new requirements, without concern for the pre-existing code that provides the glue into the framework. A more modular and enhanced version of OneSAF was recently released and testing will show if this new version meets the intent of this article.

PERSISTENT M&S DATA ISSUES

Even if the M&S community can agree on the concept of a common modeling framework for composing simulation, there still exist a number of data issues that need to be resolved. According to McGroder and Lashlee (2011), each community that uses models and simulation collects or generates their own required data and then pays to have the data transformed to ingest into the model or simulation in use. This means that often the same data is collected or generated multiple times by the separate communities but cannot be reused by other communities.

Generating simulation-ready data from disparate datasets takes considerable time and money, and is on the critical path to successful M&S execution (McGroder & Lashlee, 2011). Often, the data is not shared, not enhanced, and not correlated. And once the data is formatted and used, whether for a specific experiment, operational test, or training event, it is not made available for reuse across the communities. Data for training systems development currently require a significant level of effort to collect and format for use. Non-system, high-cost visualization data (i.e., to support 3-D models, sensor data, and terrain databases) is an expensive but necessary category of data used by modelers in the systems life cycle (Hartman et al). These data are particularly important for testing, concept development, and training

Hartman et al (2014) found that there is no uniform authoritative taxonomy that spans all categories of acquisition data. Some categories of acquisition data have de facto, consensus-based or mandated taxonomies for data within those categories. These primarily include physical characteristics data for products/systems (although there are multiple categorizations, not always consistent with one another), manufacturing process data, and operations and support cost data (to perhaps the third level of detail, after which there are variations by system type). System performance data appears to be the category where there is the least degree of accepted hierarchical categorization of data. In particular, data for training systems development require a significant level of effort to collect and format for use.

The Government often encounters challenges with collecting data from the system developer due to proprietary concerns, lack of data maturity, lack of specificity for what data are required, and administrative delays (Hartman et al, 2014). Contractual issues play an important role in the lack of availability of data for training and analysis, with proprietary restrictions remaining one of the foremost concerns. Review of many programs revealed that contract data packages are inconsistent or lack specificity in and across programs—even within system classes and Services.
Variability in types of procurements and within the Service program offices make standardized data structures difficult to achieve and has resulted in a number of competing data taxonomies and blue force data generation processes (Hartman et al, 2014). Systems data evolve as the system experiences periodic updates through the life cycle.

CORRELATING SYSTEMS DATA THROUGHOUT THE ACQUISITION LIFECYCLE

The key is to avoid paying for systems data more than one and provide a mechanism for consistency in representation of the system across the lifecycle (including support for experimentation, test and evaluation, analysis, and training) is to collect, correlate, with traceability, systems data as the system matures through the phases of the lifecycle. This does not mean system data will remain the same as the system matures. Rather, the data should be organized in a system of system infrastructure, such that traceability to system requirements is maintained (DoD SE Strategy).

For example, Blackburn, Bone & Witus (2015) developed an integrated framework, described as ‘a more holistic model-centric engineering approach’, for Naval Air Systems Command (NAVAIR). The expected capability of such an approach would enable mission-based analysis and engineering that ‘reduces the typical time by at least 25 percent from what is achieved today for large-scale air vehicle systems.’ The effort investigates the technical feasibility of moving to a “complete” model-centric lifecycle with traceability from requirements to system design and performance. The proposed model-centric engineering approach has three key critical technical items: 1) cross-domain and multi-physics model integration, 2) ensuring model integrity (trust in the model predictions), and 3) high performance computing, which is an enabler for 1 and 2, but critical due to the scale and complexity of next generation systems.

MBSE AND SIMULATION

Secretary of Defense directed the adoption of new practices to achieve greater performance and affordability to meet current and future challenges. The DoD Digital Engineering Strategy outlines the Department's five strategic goals for the digital engineering initiative (DoD, 2019). The goals promote the use of digital representations of systems and components and the use of digital artifacts. The first goal is to formalize the development, integration, and use of models to inform enterprise and program decision making.

Digital technologies have revolutionized business across most major industries, and our personal life activities. Through increased computing speed, storage capacity and processing capabilities, digital engineering has empowered a paradigm shift from the traditional design-build-test methodology to a model-analyze-build methodology. This approach is anticipated to enable DoD programs to prototype, experiment, and test decisions and solutions in a virtual environment before they are delivered to the warfighter. This transition to Digital Engineering also provides an opportunity to address some of the M&S data issues.

According to Blair et al (2018), model-based systems engineering (MBSE) is being utilized in a number of enterprise applications to manage large, complex, long life-cycle systems such as ballistic missile defense systems. Additionally, modeling and simulations are heavily employed to support performance assessment throughout a system's life cycle, with an emphasis on utilizing M&S to aid and reduce cost within development, integration, and test. MBSE is commonly applied early in the systems engineering process to support requirement derivation and architecture definition (Blair et al, 2018). Modeling tools are applied to create a behavioral (also referred to as functional) representation of the system. These tools leverage the systems modeling language (SysML) to identify and present the sequence of tasks the system performs. A key recent objective has been to combine M&S with MBSE in order to 1) enhance development and evaluation of performance requirements alongside traditional methods of deriving functional and behavioral requirements 2) connect M&S and MBSE utilities to allow integrated comprehensive requirements development rather than parallel, disconnected paths for performance and functional requirements and 3) ensure M&S is implemented such that the M&S tools utilized to support requirements development are compatible with or extensible into the higher fidelity physics-based, engineering-level M&S tools utilized as system maturity develops.

Blair, Sholes & Reed (2018) describes a new methodology for creating system behavioral simulations in tandem with existing MBSE processes being developed within the U. S. Army Aviation and Missile Research Development and Engineering Center (AMRDEC). The data-driven approach utilizes system functional definitions, event sequences, and transition data from existing MBSE tools such as IBM Rhapsody to dynamically build a behavioral simulation from a library of component math, physics, and behavior models. This approach significantly reduces the integration effort required to create or update a simulation to reflect incremental changes to the system. The proposed methodology maintains traceability of simulation behaviors to their underlying requirements, allowing a simulation to evaluate requirements' impact on the system's behavior and assess sensitivity, suitability, and consistency of those requirements with respect to each other and the system's simulated performance. The resulting simulation can be executed independently of the original MBSE environment, enabling new use cases for systems engineering models, such as directly supporting integration into discrete event simulation frameworks and high-run-count Monte Carlo scenario execution.

AMRDEC approach towards developing a unified concept for MBSE and M&S is to begin with a MBSE model and identify what extensions allow for its use as a simulation (Blair et al, 2018). According to Blair et al, a MBSE model defines a body of system functions, states, or behaviors, and their times, transitions, and relations to each other. If one was to start a clock and select some initial states for the system, one could follow along the sequences of activities defined within the MBSE model in simulated time. AMRDEC developed an executable simulation framework, called STRIDE, for this purpose. STRIDE captures the MBSE modeling and integrates various algorithms to provide an integrated behavioral representation of the system, as seen in the figure below. A more detailed architecture can be found in Blair et al (2018).

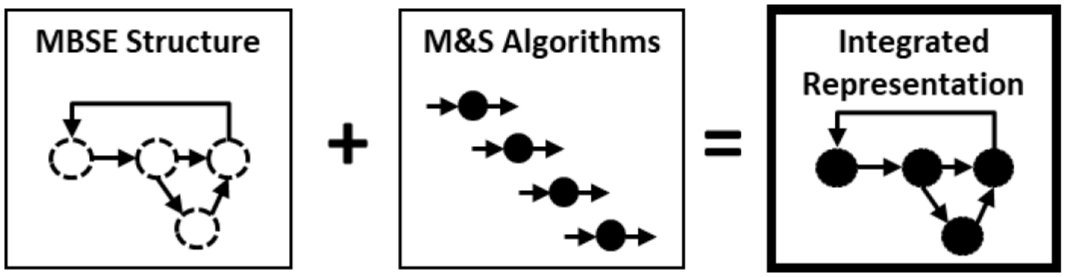

Figure 1. MBSE and Simulation Relationship (from Blair et al, 2018)

This executable simulation framework presents the first step in bridging the MBSE environment with a constructive simulation of the operating environment used by the broader M&S community for analysis, experimentation, operational test and evaluation, and training. This behavioral execution of the MBSE system model is not yet a simulation because, it does not provide for simulating all the other entities and interactions in an operational environment. For that, we need to capture the relevant modeling and data that enables representing the system performance and behavior in a constructive simulation environment.

The AMRDEC executable simulation framework can be used to generate system performance and behavior data needed for constructive simulation representation of the system in an operational environment to validate effects on overall mission effectiveness and support new operating concepts. A formalized infrastructure and process for deriving the performance and behavior data from MBSE-based modeling for ingestion into battlespace constructive simulations is required to enable reuse and cost savings.

FUTURE PLANS

AMRDEC executable simulation framework, STRIDE, can generate performance and behavior data that should be able to be ingested into platform models in a modeling framework, like AFSIM, and simulated in a scenario representative of the conceptual battlespace for that system. Our intent is to capture performance and behavior data from STRIDE and ingest into a model in AFSIM to quickly compose a simulation to assess the system performance in an operationally representative scenario. The reason for selecting AFSIM is due to the large number of user organizations (over 200 announced at recent AFSIM Users Group meeting) across the Air Force, Navy, and Army and preexisting modeling done for scenarios relevant to the system in development by AMRDEC. This test will be used to demonstrate the potential for a more rapid mission effectiveness assessment of a new system during design and development, as well as facilitate development of tactics, techniques and procedures associated with integrating the new system into the total force structure.

In addition to facilitating capture of system data from MBSE processes, another key to rapidly developing agile simulation capabilities will be a common modeling framework, below the federate level, in which newly developed technologies and tools from industry, academia, and DoD labs can be integrated and tested; and then retained in a persistent modeling environment. The advantage of this new approach is that this new synthetic environment will provide needed capabilities faster; then validated by military operators. This new modeling framework will also enable sharing of investments and facilitate reuse. This approach will also set the stage for capture and reuse of systems data generated during design and development of future systems.

Keeping in mind this desired endstate, each of the highlighted modeling frameworks has emerged with a focus on particular domains. For example, while AFSIM is designed to enable modeling of air, land, maritime, space and cyber domains, the majority of modeling to date has been focused on combat air missions, with little ground maneuver. Even though Army and Navy have begun using AFSIM, there is still relatively little modeling of, for example, ground assets. Whereas, OneSAF is primarily modeling assets associated with the land domain. Therefore, work is planned to test compatibility between AFSIM and OneSAF (both coded in C++) to leverage the modeling in each for a better representation of multi-domain operations. Part of the testing will address whether AFSIM can use OneSAF modeling directly or require federating the two in a simulation scenario through APIs or DIS interface. This will provide insight into how difficult migration to a single modeling framework for DoD will be to achieve.

ACKNOWLEDGEMENTS

This paper would not have been possible without the willingness of many dedicated experts across DoD to share experiences and insights.

REFERENCES

Advanced Framework for Simulation, Integration and Modeling (AFSIM) Version 2.0 Overview, August 2016 Technical Report, Air Force Research Laboratory.

Akbas, A. & Farwowski, W. (2015). A MBSE Approach in Modeling Systems Using Hybrid Simulation Techniques. *The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, 2015.

Bartholet, R. Brogan, D. Reynolds, P. & Carnahan, J. (2004). In Search of the Philosophers Stone: Simulation Composability versus Component-Based Software Design. *Proceedings of the 2004 Fall Simulation Interoperability Workshop*, Orlando, FL 2004.

Billings, S. et al (2016). Joint Training Implementation of Modeling and Simulation (M&S) as a Service (MSaaS). *The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, 2016.

Blackburn, M., & Verma, D. (2017). Transforming Systems Engineering through Model-Centric Engineering. SERC-2017-TR-110, 2017.

Blair, M., Sholes, E., & Reed, G. (2018). A Methodology for Componentized Simulation Generation Using MBSE Definitions. *The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, 2018.

Clive, P., Johnson, J., Moss, M., Zeh, J., Birkmire, B. & Hodson, D. (2015). Advanced Framework for Simulation, Integration, and Modeling (AFSIM). *International Conference of Scientific Computing*, 2015, 73-77.

Culton, T., Parkes, D., & Walrond, T. (2016). STO-MP-MSG-143-18 Future Construct/Architecture for Modeling and Simulation Support to Joint and Collective Training Across the Continuum of Military Operations.

Department of Defense Digital Engineering Strategy 2018. Available at: https://www.acq.osd.mil/se.

Ford, K. (2018). Modelling & Simulation as a Service – Empowering Operational Users. *The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, December, 2018.

Gaughan, C. Metevier, C., Athmer, K., Gallant, S., Murphy, S., & Snively, K. Bringing Next Generation Simulation into the Land of Practicality. *Proceedings of the Fall Simulation Interoperability Workshop*, 2013.

Hannay, J. & van den Berg, T. (2014). The NATO MSG-136 Reference Architecture for M&S as a Service. STO-MP-MSG-149.

Hartman, F., Coolahan, J., Hughes, A., Sanders, C., & Wisher, R. (2014). Model Applications Across Acquisition Program Life Cycle: Highlighting Data for Systems Engineering, Training, and Test. IDA Document D-5295, September 2014.

Kasputis, S. & Ng, H. (2000). Composable Simulations. *Proceedings of the 2000 Winter Simulation Conference*.

Kewley, R. & Sapol, S. (2014). Executable Architecture for Systems Engineering – Distributed Modeling Framework. *Proceedings of the Simulation Interoperability Workshop*, 2014.

McGroder, D. & Lashlee, J. (2011). An Emerging DoD M&S Enterprise Data Strategy. *Simulation Interoperability Workshop*, Orlando, FL, 11S-SIW-029.

NATO MSG-136

Sanders, C. (2018). A Call for a New Army Modeling Framework. *Simulation Innovation Workshop*, Orlando, FL, 18W-SIW-031.

Sarjoughian, H., Kim, S., Ramaswamy, M., & Yau, S. (2008). A Simulation Framework for Service-Oriented Computing Systems. *Proceedings of the 2008 Winter Simulation Conference*. 845-853.

Siegfried, R., van den Berg, T., & Cramp, A., Huiskamp, W. (2014). , M&S as a Service: Expectations and challenges. *2014 Fall Simulation Interoperability Workshop*, pp. 248-257, Orlando, FL.

Siegfried, R., & van den Berg, T. (2015). M&S as a Service: Paradigm for Future Simulation Environments. *The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL, December, 2018.

Snively, K., Leslie, R., & Gaughan, C. (2013). Runtime Execution Management of Distributed Simulations. *Proceedings of the 2013 Winter Simulation Conference*.

Tolk, A., C. D. Turnitsa, S. Y. Diallo, and L. S. Winters. (2006). Composable M&S web services for net-centric applications. The Journal of Defense Modeling and Simulation, 3(1): 27-44.

Tolk, A. & Mittal, S. (2014). A Necessary Paradigm Change to Enable Composable Cloud-based M&S Services. Proceedings of the 2014 Winter Simulation Conference: 356-366.

Winfrey, C. M., Baldwin, B. A., Cummings, M. A., Ghosh, P. (2014) OSM: an evolutionary system of systems framework for modeling and simulation. *Proceedings of the 2014 Annual Simulation Symposium*: p. 1.

Zeigler, B., Kim, T., & Praehofer, H. (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, Second Edition: Academic Press.