Bringing Next Generation Simulation into the Land of Practicality

Chris Gaughan & Christopher J. Metevier
Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC)
Orlando, Florida
chris.gaughan@us.army.mil
chris.metevier@us.army.mil

Keith Athmer
Training and Doctrine Command (TRADOC) Maneuver Support Center of Excellence (MSCoE)
Fort Leonard Wood, Missouri
keith.e.athmer.civ@mail.mil

Scott Gallant
Effective Applications Corporation
Orlando, Florida
scott@effectiveapplications.com

Shaun Murphy & Keith Snively
Dynamic Animation Systems, Inc.
Orlando, Florida
smurphy@d-a-s.com & ksnively@d-a-s.com

ABSTRACT

Guess what? You've got a simulation experiment to do. It supports Force Protection. It is important to your leadership in the Training and Doctrine Command (TRADOC). Here are two ways that this could play out…

Situation 1: You did a previous Force Protection experiment last year that has elements of what you want. You look at your support staff and realize that there has been turnover from last year so you do not have the same experts who ran that experiment. Furthermore, some of the machines that ran the experiment have been repurposed. You are nearly starting from scratch. You also have the terrible epiphany that you could be in this boat again the next time you do a Force Protection experiment if conditions stay the same.

Situation 2: You did a previous Force Protection experiment last year that has elements of what you want. The good news is that you captured the knowledge of how the simulations work into encapsulated functional "building blocks." These building blocks can then be put together automatically based on their interface compatibility as required for simulation usage without having to start from scratch. Moreover, you have an interface to allow users to specify scenarios, capabilities, etc., required for their purpose. The fact that you have had turnover in your support team no longer matters thanks to this system. Finally, you have access to a deploy asset manager that can push machine configurations out dynamically to any assets that you want so the changing state of hardware is also not an issue.

This paper discusses how we have taken advantage of U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC) research that makes Situation 2 a reality, focusing on a demonstration of these technologies with the TRADOC Maneuver Support Center of Excellence (MSCoE) in support of Force Protection. To this end, we will discuss a system that is configurable, deployable, functionally appropriate and repeatable, saving hardware and labor costs.

ABOUT THE AUTHORS

Chris Gaughan is the Chief Engineer for Advanced Simulation and Deputy Technology Program Manager of the Modeling Architecture for Technology, Research and EXperimentation (MATREX) program at the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC). He has a diverse portfolio of distributed simulation projects that support the full spectrum of the Department of Defense Acquisition Life Cycle. He received his Master of Science and Bachelor of Science in Electrical Engineering from Drexel University in Philadelphia, PA.

Christopher J. Metevier is the Chief of the Advanced Simulation Branch and Technology Program Manager of the MATREX program at the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate

(HRED) Simulation and Training Technology Center (STTC). He has over 23 years of experience with the Army and Navy in the Modeling & Simulation (M&S) field. His M&S experience extends across the acquisition lifecycle and includes the research, development, adaptation, integration, experimentation, test and fielding of numerous simulation technologies and systems. He received his Master of Business Administration from Webster University and his Bachelor of Science in Electrical Engineering from the University of Central Florida.

Keith Athmer is a Computer Scientist at the Training and Doctrine Command (TRADOC) Maneuver Support Battle Lab (MSBL) in Fort Leonard Wood, MO. He has been with the MSBL since July 2005 where he serves as the Modeling & Simulation (M&S) Systems Engineer/Architect, M&S Project Lead and Application Administrator for the MSBL simulation network. He provides M&S expertise and technical support for large-scale multi-lab TRADOC events. He holds a Bachelor of Science in Management Systems with a Minor in Computer Science from the University of Missouri - Rolla.

Scott Gallant is a Systems Architect with Effective Applications Corporation. With over eighteen years of professional experience integrating distributed systems, Mr. Gallant has concentrated his focus towards improving system of systems interoperability through systems engineering processes, tools and best practices. He helped create and develop a data-driven systems engineering tool for the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC) along with the Executable Architecture Systems Engineering (EASE) project described herein. Mr. Gallant earned his Computer Science degree from George Mason University.

Shaun Murphy is an accomplished software engineer who has held research, development and productization positions in simulation and training infrastructure services and government communications companies for the past decade. He has worked several project domains including mixed fidelity deployment environments, synthetic environment services, battlefield management systems and interoperability systems. Previously, Mr. Murphy held an associate research position with the Institute for Simulation and Training at the University of Central Florida for embedded simulation and training. Mr. Murphy is currently the technical lead of the Executable Architecture Systems Engineering research project and is researching and developing cloud deployment services and Platform as a Service technologies for stand-alone and stand-together deployment of heterogeneous applications and virtual networks with Dynamic Animations Systems, Inc. in support of the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC).

Keith Snively is a Principal Software Engineer with Dynamic Animation Systems, Inc. He has over 15 years of experience in distributed computing and Modeling & Simulation (M&S) for the Department of Defense. He has worked on design and development of several distributed communication systems, including the Run Time Infrastructure Next Generation (RTI-NG) and the Test & Training ENabling Architecture (TENA) middleware, and also participated in the development of the High Level Architecture (HLA) 1516 Evolved specification. Currently, Mr. Snively serves as a principal software developer in support of the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC) and specifically supports design and development of the MATREX RTI-NG, ProtoCore and EASE. Mr. Snively received an M.S. degree in Mathematics from the University of Virginia.

Bringing Next Generation Simulation into the Land of Practicality

Chris Gaughan & Christopher J. Metevier
Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC)
Orlando, Florida
chris.gaughan@us.army.mil
chris.metevier@us.army.mil

Keith Athmer
Training and Doctrine Command (TRADOC) Maneuver Support Center of Excellence (MSCoE)
Fort Leonard Wood, Missouri
keith.e.athmer.civ@mail.mil

Scott Gallant
Effective Applications Corporation
Orlando, Florida
Scott@EffectiveApplications.com

Shaun Murphy & Keith Snively
Dynamic Animation Systems, Inc.
Orlando, Florida
smurphy@d-a-s.com & ksnively@d-a-s.com

BACKGROUND

Modeling & Simulation Is Good For You But That Doesn't Make It Taste Any Better

The argument has been made and continues to be made for why we need Modeling and Simulation (M&S) in the Department of Defense (DoD). Use cases include experimentation, acquisition and training, to name a few. In each of these cases, simulation is best used when the phenomena represented within the simulation is time consuming, difficult, dangerous or expensive to perform live. Simulations built within DoD can be reused for many organizations and purposes. This reuse of simulation provides cost savings but it also introduces interoperability issues and engineer knowledge retention issues.

In general, M&S are developed for a specific need within a limited scope and funding. While reuse is definitely encouraged in order to save money and time, each use case is unique and has specific modeling and simulation needs. Changes must be made to the reused M&S assets in order to meet the new scope and requirements. These changes can be difficult due to rapid technical staff turnover and the complex details of M&S. Changes that managers think are trivial can frequently be very time-consuming and risky.

Should These Things Work Together?

The concept of simulation interoperability is always an interesting topic. What seems so simple, specifically simulations working together towards a common goal, in reality is quite nontrivial. The literature has discussed what it means to be interoperable (Tolk and Muguira, 2003; Gaughan and Gallant, 2011) and how it is deceptively difficult due to the disjointed means of interoperability; e.g. object models, middleware, semantic understating, etc. Simulation interoperability is an issue because it inherently includes a number of software applications that represent a key portion of the domain under study and all of those applications were built by a different organization with a different purpose. Those models and simulations have varying technical attributes and have varying resolution of detail for their respective representation of the battle space. Getting all the models to work on the same technical simulation middleware is just one aspect of interoperability. Time must also be spent getting the applications to use the same object model (syntax) when communicating over the network and middleware. Even after the applications are using the same object model, the semantics of those communications must be synchronized. The reasoning behind the interactions is not always the same between two models.

When considering multiple models and/or simulations working together, configuration of the applications to implement a scenario properly, communicate on a common technical plane and represent the phenomena consistently with the other applications can become very complicated. This configuration may change as the M&S architecture changes or the scenario changes. It further introduces a requirement that the engineering staff working with the application be trained and become especially knowledgeable on each application.

Where Did Everybody Go And Who Mucked With My Lab?

Staff turnover becomes an important issue due to this kind of specialized knowledge. Simulationists may find a new job or a new contractor may win the support contract. M&S within DoD is a very niche domain so the pool of knowledgeable simulationists is small. Also, due to the complexity of the simulations and the configurations required for specific scenarios and architectures, even the best engineers will have trouble retaining knowledge of exactly how to replicate what they did just a few years ago. This means that once an experiment is concluded, it becomes increasingly difficult over time to execute the simulation environments from the past.

Lab assets including hardware, software and networks need to be constantly upgraded and maintained. This information technology function is time-consuming, erroneous and expensive. Every time there are software updates from the DoD program that develops each one of the simulations, all of the previously mentioned issues are exasperated.

I See Dead Simulations

The development cycle of DoD M&S truly varies from application to application. In many cases, robust documentation, testing and Verification & Validation (V&V) is not possible. Instead, the institutional knowledge of a particular model or simulation stays with the developer/proponent. Unfortunately, as funding and interest ebbs and flows, this institutional knowledge demonstrates how perishable it is and quite often is lost leaving an orphaned model and/or simulation that cannot be easily resurrected. This situation is systemic of the DoD environment and is very prevalent in the experimentation, analysis and acquisition communities where many models and simulations are built quickly with limited funding.

One important user of simulation that has experienced many of these difficulties associated with the use of complex simulation is the Training and Doctrine Command (TRADOC) Maneuver Support Center of Excellence (MSCOE) with their Maneuver Support Battle Laboratory (MSBL) in Fort Leonard Wood, Missouri. This paper will discuss research that attempts to alleviate some of these challenges.

WITH OUR POWERS COMBINED

Introducing The Players

TRADOC MSCoE is the Army proponent for protection and in turn, has the mission to support fixed site protection issues. To this end, the MSBL repeatedly develops M&S capabilities by integrating various models and simulations to support Force Protection Joint Experiment (FPJE) and the development of Standard Operating Procedures (SOPs), Tactics, Techniques and Procedures (TTPs) and Concepts of Operation (CONOPS).

The U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation and Training Technology Center (STTC) has the mission to conduct simulation and training technology Research and Development to enhance Warfighter effectiveness. STTC's Advanced Simulation Branch works to research and develop innovative methods and means that facilitate the design, integration and use of simulation and training technologies to enable effective training, experimentation, test and acquisition in support of the Warfighter. A key to this is reducing the cost and risk of creating and executing simulations via more efficient and effective development, integration and test.

Introducing The Use Case

In support of Force Protection experimentation, MSBL has built a Virtual Base Defense Operating Center (VBDOC) that provides a venue for the exploration, development and refinement of SOPs, TTPs and CONOPS under consideration by the U.S. Army. This VBDOC is powered by a simulation architecture containing a force-on-force simulation, a suite of Chemical, Biological, Radiological, Nuclear (CBRN) simulations, a desktop Unmanned Ground Vehicle (UGV) Advanced Concepts Research Tool (ACRT), a simulated sensor controller and simulated video feeds from various sensors. This simulation architecture stimulates Mission Command systems, such as the Joint Battlespace Command and Control System (JBC2S) and the Joint Warning and Reporting Network (JWARN).

While the VBDOC indeed has a known simulation architecture, it is very dependent on specific hardware assets, it is susceptible to changes in software versions and it requires specialized expertise to execute. As this year's Force Protection experiment looms, previous experience lends itself to the following situation: Last year's Force Protection experiment has elements of what is required to execute the requirements of the current experiment. The support staff has had turnover from last year so there are not the same experts who ran that experiment. Additionally, some of the hardware used for last year's experiment has been repurposed. In many ways, it is like starting from scratch and having to piece together the puzzle from what limited knowledge of last year still remains.

Alternate Reality

Imagine this situation instead: Last year's Force Protection experiment has elements of what is required to execute the requirements of this year's. The good news is that the knowledge of how the simulations work has been captured into encapsulated functional "building blocks." These building blocks can then be put together automatically based on their interface compatibility as required for simulation usage without having to start from scratch. Moreover, there is an interface that allows users to specify scenarios, capabilities, etc., required for the experiment. While there has been turnover in the support team, this capture has supported the new team's ability to execute. Finally, there is a deploy asset manager that pushes machine configurations out dynamically to hardware assets so the changing state of hardware is not an issue.

Making This Alternate Reality Actual Reality

The ARL HRED STTC is conducting research into this problem space through the Executable Architecture Systems Engineering (EASE) project. The goal of the project is to provide an executable architecture based on systems engineering for M&S. The approach allows users to traverse systems engineering information to compose and execute simulation scenarios that address their analysis or training goals (Gallant, Metevier and Gaughan 2011). The goal of EASE is to lower the barrier of entry to the use of M&S by providing a single interface for systems engineers, software developers, information technology professionals and analysts to work together to define the simulation systems engineering data and execute the right applications in order to support the M&S user's goals.

EXECUTABLE ARCHITECTURE SYSTEMS ENGINEERING IN CONTEXT OF THE MANUEVER SUPPORT BATTLE LABORATORY

A prototype from the EASE research project was recently deployed to MSBL to help improve how their simulations are managed and executed. The EASE architecture consists of a number of components shown in Figure 1, specifically the Software Design Description (SDD), the User Interview, the Deployment Management System (DMS) and the Coordinator.

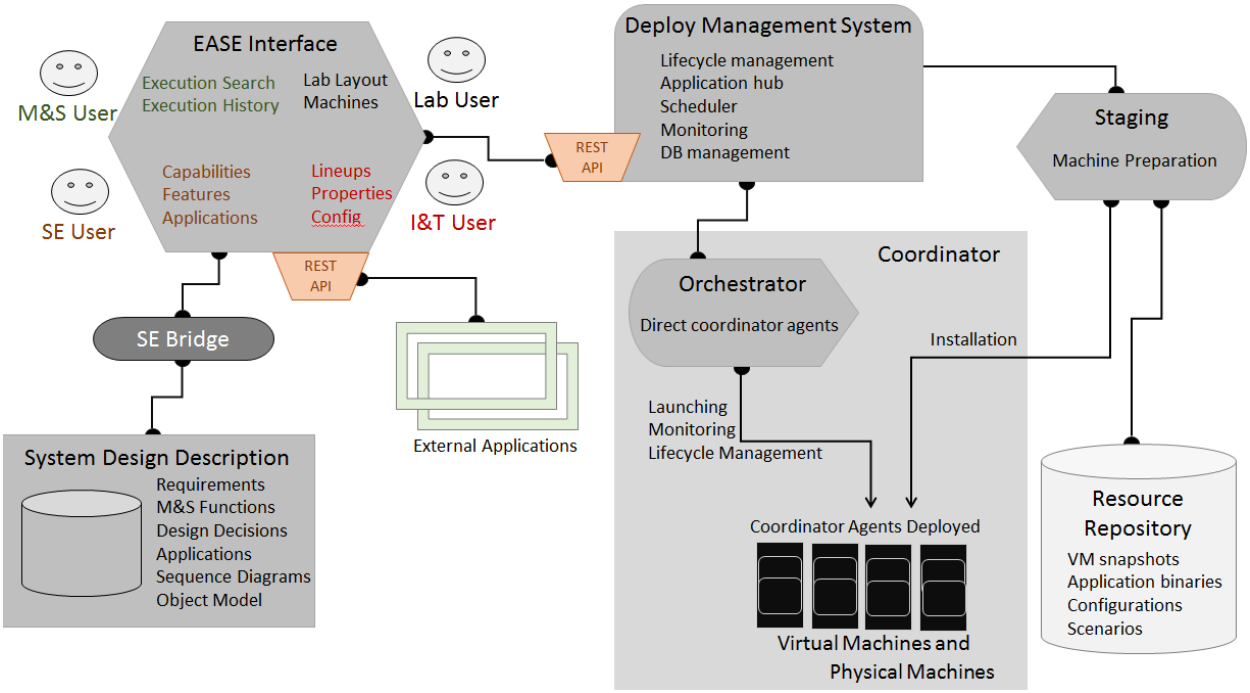

Figure 1. EASE Components

The SDD captures the systems engineering information on the available simulation applications, their capabilities and how they interoperate in a simulation environment. It also allows the system engineering user to add new simulation applications (Beauchat, *et al.* 2012).

The EASE Interview System allows the user to traverse captured systems engineering information to select and compose a simulation system. The user is presented with a list of options based upon scenario criteria and functional capabilities and has the ability to customize components of the scenario (Gallogly, et. al. 2012). Additional advanced capabilities allow the user to inject custom properties and create surrogates to fill in specific capabilities.

Once the scenario has been designed and the components chosen, The EASE DMS determines the necessary assets for execution and deploys software and configuration files. It employs Platform as a Service (PaaS) to utilize virtual and hardware assets in support of a simulation exercise (Murphy, Gaughan, Diego and Gallant 2012). Its tasking service then determines how and when to run a simulation execution.

Finally, the EASE Coordinator is responsible for the actual execution of the simulation exercise. The Coordinator handles the Time Sequence of Events provided by the tasking system and controls the launch, initialization, shutdown and cleanup of each process. The Coordinator is also responsible for progression of the overall simulation execution ensuring all processes perform the necessary tasks at the proper time.

Figure 2 shows the EASE roles in context of where they interface with the system. The M&S user accesses the system through a web interface that allows the user to explore capabilities, select desired scenarios and configure parameters for simulation – all of the complexities of simulation are abstracted away from this user. The Systems Engineer details the simulation interfaces within the SDD and links that design with capabilities – linking the technical design with capabilities gives EASE the applications that are necessary to exercise the M&S users' needs as well as identifies which key information exchange elements are critical for data collection and further analysis. The Developer creates and manages the M&S software – they have an interface into EASE to upload or point EASE to the path of their software, the configurations and how to execute the software. Finally, the Integrator connects the technical details entered by the developers and systems engineers with the simulation capabilities and scenarios that the M&S users will execute – they adjudicate application lineups and manage the user interface options for the M&S users.

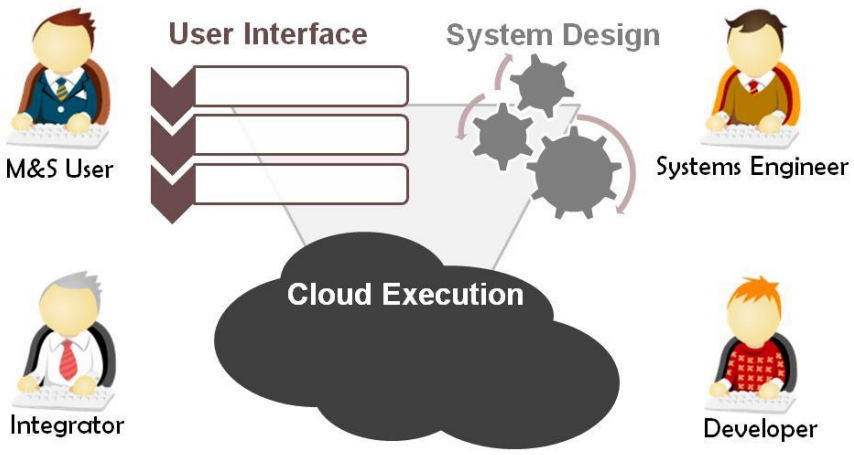

Figure 2. EASE Roles

User Interface

EASE provides an interface to M&S users to select the capabilities they need and the scenario necessary to stimulate the appropriate warfare circumstances. The selection criteria finds and avails the most appropriate executions for the user to choose from. The user can then adjust predefined configuration elements that have been exposed by the developers, select the number of runs they need to execute, schedule a run for the future and hit the "Go" button to execute. The interface, realized using web technologies, results in potentially complex M&S executing in the cloud or on identified lab machines.

The Systems Engineers, Developers and Integrators can centrally manage all aspects of EASE and how to execute the proper M&S systems to achieve the M&S users' needs. Having a data-driven and easy to use interface keeps information current. In turn, each user can be assured that they're referencing and updating the latest information. Typical M&S programs use Microsoft Office files to store and manage the details of the M&S environment. This creates a configuration management problem where staff must ensure that they have the latest version. A data-driven approach also facilitates that the information is linked within the database so it is easier to see references within a web interface rather than having to open and navigate multiple spreadsheets or word processing documents.

At MSBL, they are using a version of the EASE interface tuned to their experimentation needs. The EASE interface provides the MSBL staff a dashboard view, shown in Figure 3, of all the most critical elements for their use case including the applications installed and available, sets of applications that can execute together (line-ups), scenarios available to execute and the floor plan layouts which dictate which machines the simulations will execute on. This dashboard set of views simplified the user experience from updating rudimentary forms to update a relational database to more of a function focused set of forms to accomplish their mission. This alleviates the users from knowing the underlying data structures of EASE and keeps them focused on their mission.

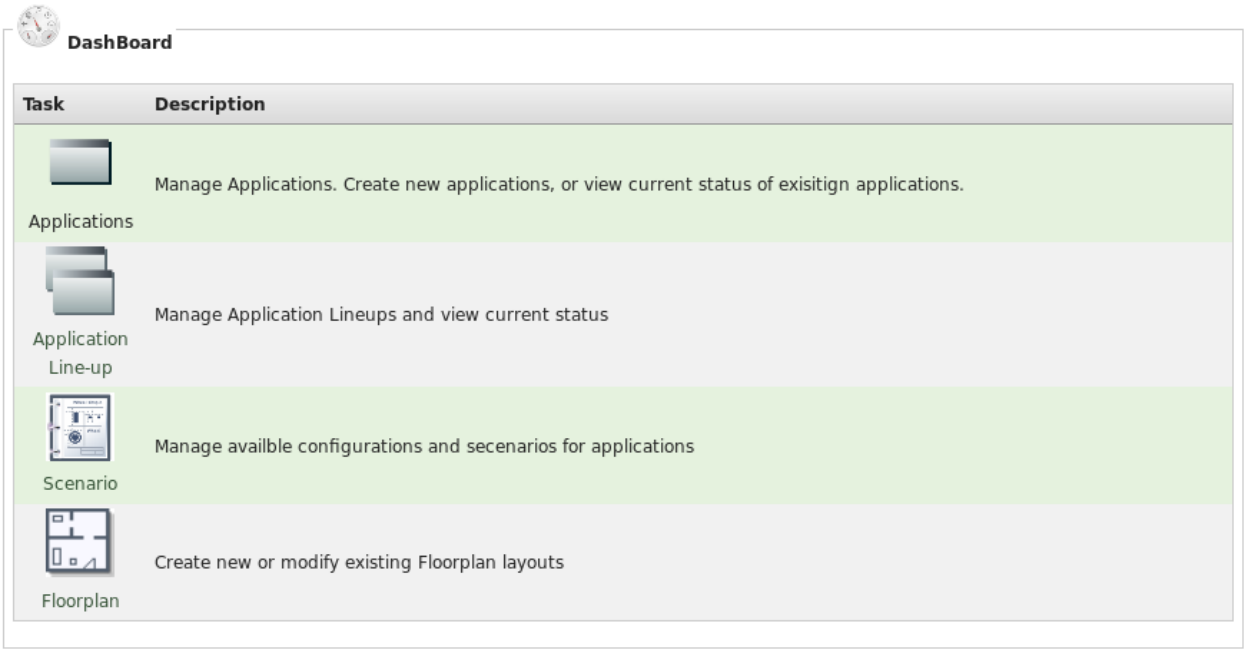

Figure 3. EASE Dashboard

Systems Engineering Linkage

When integrating models together, we must have an understanding of the functions that each model implements and their data and functional dependencies within a higher level understanding of the system. A decomposition of requirements from a semantic interoperability perspective would identify the need for an interface change early in the development lifecycle. The typical suite of models, tools and architecture allow for many different possible configurations of the system to achieve the user's functional requirements. The goal is to work with the user to develop an event-specific SDD that contains their exercise requirements, data decomposition requirements, system architecture guidelines, scenario, configuration choices and model selection. Systems Engineering for distributed systems is complex and unique from Systems Engineering for stand-alone software systems. Our Systems Engineering process assists the user by mapping their functional requirements to model behavior and mapping the interfaces to the existing object model fields as best as possible.

The simulation architecture includes many disparate models (differences with functionality, fidelities, resolutions, technical dependencies, etc.) that need to work together. These models can be combined within an architecture that scales by using complicated techniques, such as Data Distribution Management (DDM) (Van Hook and Calvin. 1998). Those technical complexities force design, development and testing collaboration to be well coordinated and as automated and data-driven as possible. Simulation requirements such as "fair fight" issues, scalability concerns and data element analysis force the design to have additional architectural strategies that must be uniformly followed. These architectural strategies must be captured and enforced with an SDD that is linked to the design. The SDD captures the system design at a functional level and subsequently links the functional design to the technical design. This allows the functional requirements to be linked to system design and allocated to specific models.

The SDD is planned for use at MSBL in the future. Once in use, EASE will allow for engineers to capture the systems engineering details of the simulations with its own data-driven SDD and link that technical design to functional capabilities to be represented within the simulation environment. As MSBL brings new simulations into their environment to support Force Protection, the linkage between technical design and capabilities allows EASE to determine which applications best interoperate and what scenarios can be supported by varying M&S architecture designs.

By using the SDD, MSBL will be able to retain key design details within a persistent data store so that staff turnover or having to remember numerous constantly changing technical details will no longer be a problem. The system

design will be linked to the execution environment such that as changes are made, their impact can be readily apparent. As new models or new versions are introduced to the MSBL, the design can also be updated within new versions of the architecture. The architecture versions will keep the distinct set of interoperable applications together while historical versions of the system design can remain intact for reference, rerun or future execution.

Applications And Their Execution Details Captured

When a user wants to run a simulation, they need applications that provide the desired capabilities for simulation. One of the requirements to run a simulation is that those applications can run on the same scenario. Scenario is an overloaded term that has lots of implications, such as terrain, entities, etc. For our purpose, scenario is a set of initial conditions that all applications in the simulation must agree to. This simplified view of scenario allows all applications that have a configuration that correlates to represent a scenario, even if those applications cannot be run together. In turn, association of application configuration into a scenario is done without regard to whether or not those applications work together. The process of determining what applications can work together is done automatically as the user searches for something to run.

MSBL users, while searching for capabilities in the interview also search for scenarios. Through this search, only line-ups that have applications that can all work on a particular scenario appear. Of course, finding the right scenario is important as well. Scenarios have metadata associated with them that provides the search criteria for finding the scenario. This criterion is created by the user who installed the scenario into EASE so that they can determine what is important about the scenario. Since the application line-ups are paired dynamically with scenarios, simply adding a new application configuration to a scenario could allow many different simulations to be run.

Running a simulation requires more than just knowing what applications to run and with what scenario. Those applications and scenarios have to be installed, configured and executed on a machine somewhere. MSBL users only see applications and scenarios that are installed on the DMS. The actual execution is handled by the DMS running all of the applications and their required infrastructure in the cloud. The cloud execution is orchestrated and managed using a Platform as a Service (PaaS) approach (Murphy, *et al.* 2011).

User-Defined Simulation Configuration

Beyond the scenario and technical configurations, an application may have some performance data configurations that the user can manage. The EASE interface exposes some of the configuration attributes of the software applications within the user interface allowing the M&S user to change simulation parameters before execution. An Application Programmers' Interface (API) exists between the DMS and the interview where developers can define which attributes are configurable by the M&S user. Each item is tied to an ability to adjust the configuration, which could be an environment variable for the machine, command line options when running the application or even changing a specific line or field within a property/configuration file.

There are some consequences for changing the configuration of an application, so the fields that are exposed to the M&S user through the interview must be carefully considered by the developers and systems engineers. Some performance data could affect how the model operates and possibly lead to failure to execute as designated within the system design. The EASE system allows each field to be designated with a type, range and allowable values. This strategy of limiting what the user can configure and the ranges with which to configure the attributes facilitates the system designers to ensure proper usage of the simulation applications.

Coordinator

The runtime execution of a complex simulations working in tandem can be a complex task. Orderly management of the startup, execution, shutdown and data collection tasks are required to support iterations of a simulation run. In addition, unanticipated error conditions must be monitored to determine when an iteration should be terminated and restarted. These elements form the basis of requirements for any approach for managing a simulation execution and are the basis of the EASE Coordinator. During the initial effort, the project focused on a core set of high level requirements:

1. The Coordinator shall provide the ability to start and initialize participating processes on the necessary host machine. The Coordinator shall allow customization of these tasks.
2. The Coordinator shall provide the ability to shutdown and terminate participating processes. The Coordinator shall allow customization of the order of these tasks.
3. The Coordinator shall provide the ability to run generic processes on the necessary host within a simulation run.
4. The Coordinator shall allow specification of a milestone sequence for execution of a simulation run.
5. The Coordinator shall detect unanticipated failure of participating processes. The Coordinator shall allow execution to terminate based upon a failure.

After coordinating with MSBL on their simulation execution, we updated the Coordinator to account for launching, managing and shutting down processes on the MSBL computing architecture. The Coordinator could no longer just be put on virtual machines in a controlled environment but needed to work on Army accredited systems and deployed to desktop machines that users interact with during simulation runs.

The coordinator uses a State Chart XML (SCXML) for the control of a simulation execution. SCXML specifies a generic state-machine execution based on Harel State Tables expressed in XML (W3C 2012). Harel State Tables allow for hierarchically nested states and include associated activities for states and transitions (Harel 1987). They also form the basis of the Unified Modeling Language (UML) State-Diagrams (OMG 2009). SCXML is an evolving standard supported by The World Wide Web Consortium (W3C). There are numerous software products which provide the execution of SCXML documents supporting a variety of programming languages. Currently, the Coordinator uses the Apache Commons SCXML library as the state machine engine.

Within the coordinator, a series of state charts are used to represent the simulation execution and each process participating in the execution. Each state chart, represented as SCXML, executes asynchronously within an SCXML engine. Custom extensions to the Apache Commons SCXML library allow for the spawning of other SCXML engines and exchanging of external events with these other SCXML engines. The events may be sent between engines within the same process or across a network. The events can be used as triggers to transition to new states, set internal data or convey error conditions. Figure 4 depicts the execution of the state machines within the Coordinator architecture.

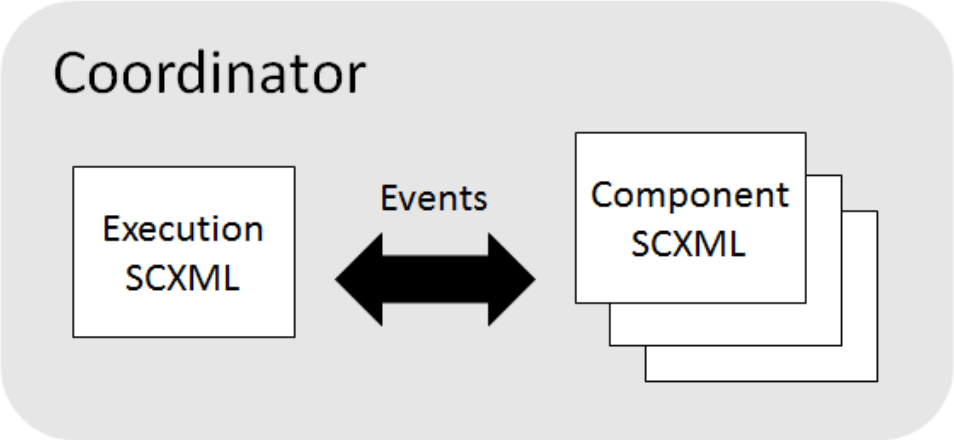

Figure 4. Coordinator SCXML engines

As depicted, the overall execution of the simulation scenario is represented as an SCXML document. The execution SCXML specifies the available states for the execution and the criteria for transitioning between the states. The execution SCXML engine sends events to and receives events from the components SCXML engines. These event exchanges progress the execution and components through their respective states. Error events may also be sent and handled as appropriate for the execution and error condition encountered. Figure 5 shows a high level sample execution where high-level states are provided.

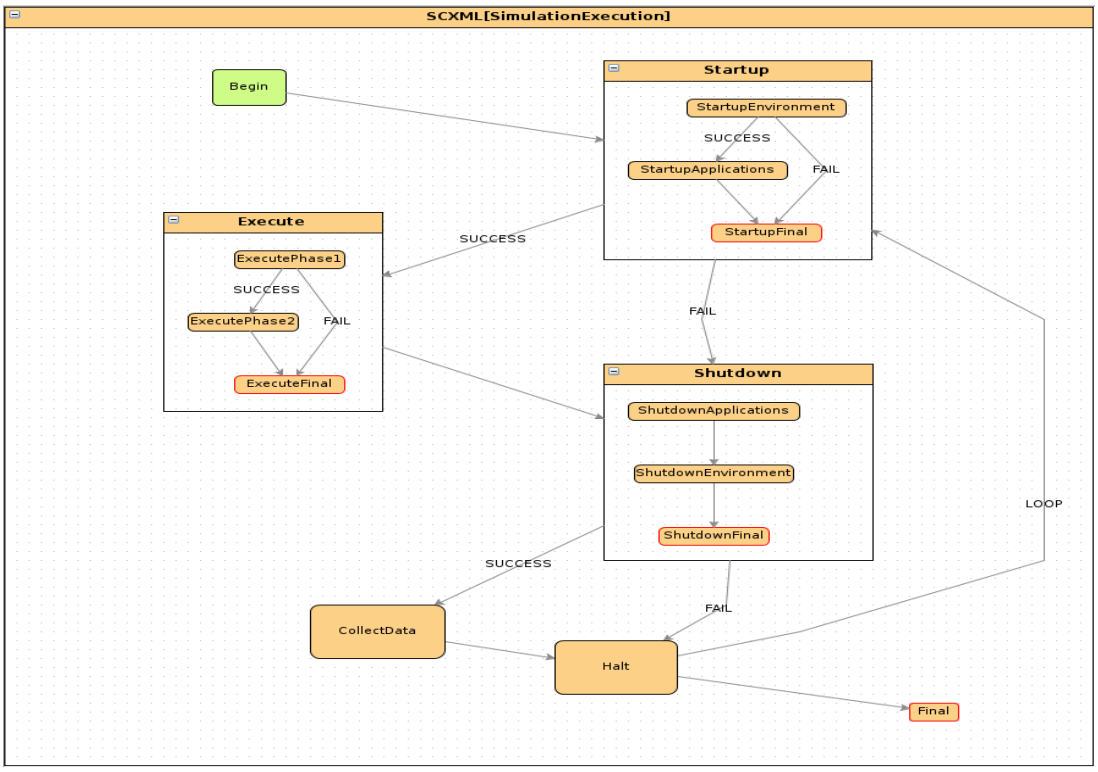

Figure 5. Simulation execution SCXML

A number of the states are composite, namely Startup, Execute, Shutdown, CollectData and Halt. Each of these composite states invokes a sub-SCXML engine that executes the behavior for that state. These sub-SCXML engines run to completion before the state under which they are invoked transitions, much like a composite state. The customizable behavior given in the coordinator XML for the simulation execution lies within these sub-SCXML engines. Using a sub-SCXML specification simplifies the respective files.

THE PROOF IS IN THE PUDDING

MSBL and ARL HRED STTC have a long standing working relationship. The integration of EASE into the MSBL is a testament to that relationship. MSBL is currently planning a host of experiments focused on aspects of Force Protection, such as entry control, CBRN defense and examining the ramifications of the proposed Army of the future. EASE will be at the center of this experimentation supporting all simulation execution but perhaps most importantly, EASE will allow MSBL to more easily reuse previous simulation environments for future experimentation while faced with a fiscal environment that definitely means reductions in staff and support. During these experiments, ARL HRED STTC will be collecting data to quantify the benefits of EASE, refining the many facets of the EASE prototype based on emerging experimentation needs and maturing EASE for transition through this usage in a relevant environment where issues such as scalability, security constraints, etc. can be explored. Through this collaboration, the combined team is bringing next generation simulation into the land of practicality through relevant use.

SUMMARY

Executing complex simulation at the MSBL will be improved by the EASE prototype that has been deployed to their lab. This is a great case study for an Army research investment to support other organizations within the Army. EASE will expedite the execution of simulation through automation, provide persistent knowledge of the complex technical details and be a tool to not only execute simulation, but also improve interoperability and automation of the execution process.

ACKNOWLEDGEMENTS

The authors would like to acknowledge ARL HRED STTC management for their guidance and direction with the maturation of the EASE research project. We'd also like to thank the engineering staff that worked so hard in the development of EASE and the MSBL staff who provided the venue for our research and worked so hard to let a Science and Technology (S&T) research project support a practical TRADOC need. Finally, the authors would like to acknowledge staff at the US Military Academy Operations Research Center for their analysis of the EASE use case and thorough examination of the DoD M&S domain for requirement prioritization and focus.

REFERENCES

Beauchat, T., Gaughan, C., Metevier, C. and Gallant, S. (2012). A Collaborative Tool for Capturing the Design of a Distributed Simulation Architecture for Composable Execution. *Spring Simulation Interoperability Workshop*. 12S-SIW-029.

Draper, N.R. and Smith, H. (1998) Applied Regression Analysis Wiley Series in Probability and Statistics.

P. DuBois. (2008). MySQL (4th Edition*)*, Addison-Wesley Professional.

Gallant, S., Metevier, C., Gaughan C. (2011). Systems Engineering an Executable Architecture for M&S. *Fall Simulation Interoperability Workshop*. 11F-SIW-033.

Gallogly, J., Gaughan, C., Marshall, H., Gallant, S. (2012). Programmatically Identifying Composable Simulation Capabilities. *Spring Simulation Interoperability Workshop*. 12S-SIW-030.

Gaughan, C., & Gallant, S. (2011). Difficulties with True Interoperability in Modeling & Simulation. *Proceedings of the 2011 Winter Simulation Conference.*

Institute of Electrical and Electronics Engineers. *IEEE Standard 1516.2-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification*. New York, NY: Institute of Electrical and Electronics Engineers, 2000.

M. Jamshidi. (2008). System of Systems Engineering. 1st ed. Wiley.

Mordecai Avriel (2003). Nonlinear Programming: Analysis and Methods. Dover Publishing.

Murphy, S., Diego, M., Gallant, S. (2011). U.S. Army Modeling and Simulation Executable Architecture Deployment Cloud Using Virtualization Technologies to Provide an Extensible Platform as a Service (PaaS) Cloud for Federated Application Configuration and Execution. *Fall Simulation Interoperability Workshop, 11F-SIW-051*

Murphy, S., Gaughan, C., Diego, M., Gallant, S. (2012). U.S. Army Modeling and Simulation Executable Architecture Deployment Cloud Virtualization Strategy. *The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.*

Powell, Edward T.. *et al.*, "The TENA Architecture Reference Document", http://www.tena-sda.org/documents/tena2002.pdf, The Foundation Initiative 2010 Program Office, 2002.

Tolk, A., & Muguira, J. A. (2003, September). The levels of conceptual interoperability model. In Proceedings of the 2003 Fall Simulation Interoperability Workshop (No. 03F-SIW, p. 007).

Van Hook, D. and Calvin, J. (1998). Data Distribution Management in RTI 1.3. In Proceedings of the 1998 Spring Simulation Interoperability Workshop: 98S-SIW-206.