Distributed Soldier Representation: Improving M&S Representation of the Soldier

Manuel Diego, Clayton W. Burford
US Army RDECOM Army Research Laboratory
Orlando, Florida
manuel.r.diego.civ@mail.mil, clayton.w.burford.civ@mail.mil

Joseph S. McDonnell, Ph.D., Bert Davis, Gary Smith
Dynamic Animation Systems, Inc.
Fairfax, Viginia
joe.mcdonnell@d-a-s.com, bdavis@d-a-s.com, gsmith@d-a-s.com

Derrick Franceschini
StackFrame, LLC
Sanford, Florida
derrick@stackframe.com

ABSTRACT

The Army has developed a breadth of Modeling and Simulation (M&S) capabilities representing platforms such as fixed and rotary-winged aircraft, tracked and wheeled vehicles, and weapons systems for various uses and of various fidelities. The Army has represented humans – soldiers, civilians, and threats – in its M&S as well. These representations provide physical model characteristics for mobility, delivery accuracy, lethality, and sensing, as well as behavioral representation to support tactical movement, clearing a building, obtaining Human Intelligence (HUMINT), and treating simulated wounded. These models rarely represent the soldier as a complex system, representing factors such as stress, human physiology, leadership, unit cohesion, and morale, to name a few. Instead, the actions of the simulated soldier are often based on a deterministic model of human behavior. When nondeterministic representations are used, they are often stochastic where random numbers provide variability across iterations, but the variability comes from the random number seed, not from the model. This provides unsatisfactory simulation results for those stakeholders interested in analyzing the effects of the soldier representation, as the simulated soldiers appear robotic or even superhuman. This paper describes the Distributed Soldier Representation (DSR) research and development effort that has been underway at the Army Research Laboratory, Human Research and Engineering Directorate, Simulation and Training Technology Center (ARL HRED STTC) for the past two years. In this paper, we describe our research that has identified eleven areas of interest for improving soldier representation. We further describe the development of an innovative Service Oriented Architecture (SOA) that provides a modern web services-based approach to integrate disparate models to address these identified representation gaps. We describe the challenges and benefits achieved by taking a web-services approach, as well as the lessons learned from the web-services integration of the Effects of Stress with One Semi-Automated Forces (OneSAF). Finally, we discuss ongoing development work.

ABOUT THE AUTHORS

Manuel R. Diego is a Science and Technology Manager within the Advanced Simulation Branch of the Simulation & Training Technology Center (STTC), United States Army Research Development and Engineering Command Army Research Laboratory (ARL HRED STTC). Mr. Diego was with the Navy Human Performance Command and now with the ARL HRED STTC managing research activities in Human Performance within a distributed Modeling and Simulation (M&S) environment.

Clayton W. Burford is a Science and Technology Manager within the Advanced Simulation Branch of the Simulation & Training Technology Center (STTC), United States Army Research Development and Engineering Command Army Research Laboratory (ARL HRED STTC). He provides technical and programmatic expertise in support of the Army Research Laboratory Human Research and Engineering Directorate Simulation & Training Technology Center (STTC) Advanced Distributed Simulation research portfolio.

Joseph S. McDonnell, Ph.D. is a Principal Scientist and Director of Modeling and Simulation with Dynamic Animation Systems, Inc. He has over 20 years experience with modeling and simulation supporting various United States (US) Government programs working in technical, staff and managerial positions. Currently, he supports Science and Technology programs at the US Army Research Laboratory Human Research and Engineering Directorate Simulation and Training Technology Center as a Principal Scientist. Dr. McDonnell received his Ph.D. in Mathematics from the University of Virginia.

Bert Davis is a Senior Integration and Test Engineer with Dynamic Animation Systems, Inc. He has over 10 years experience with modeling and simulation in support of Department of Defense simulation programs working as an integration & test lead, systems engineer, and system administrator. Currently, he supports Science and Technology programs at the US RDECOM ARL HRED STTC as a Senior Integration and Test Engineer. Mr. Davis received his BS in Electrical Engineering from North Carolina Agricultural and Technical State University.

Gary Smith is a Principal Software Engineer with Dynamic Animation Systems, Inc. He has over 13 years experience with modeling and simulation in support of Department of Defense simulation programs working as a software technical lead, software/systems engineer, and software developer. Currently, he supports Science and Technology programs at the US RDECOM ARL HRED STTC as a Principal Software Engineer. Mr. Smith received his BA in Mathematics/Computer Science from St. Mary's College of Maryland.

Derrick Franceschini is Vice President and Senior Systems Architect at StackFrame, LLC. Mr. Franceschini has over 20 years experience in the M&S industry. He has extensive experience in the Constructive and Virtual domains and is sought after in the industry for his expertise with architectural innovations for the U.S. Army's OneSAF Product Line. He is the leading force behind the conceptualization and development of the WebMSDE, a web-based Scenario Development Environment, WebSAF, a thin client, web browser-based application to visualize and control the Army's OneSAF Simulation, WebAAR, a modern, efficient web browser based After Action Review (AAR) system and the MobileSimCenter, a set of innovative products to provide easily deployable Constructive simulation capabilities. Mr. Franceschini served as the OneSAF System Architect from 2006 – 2010 and has held key development positions on the program since 2001. Earlier in his career, he served as the Lead Engineer on the Modular SAF (ModSAF) and OneSAF Testbed Baseline programs. He received his B.S. in Computer Engineering from UCF.

Distributed Soldier Representation: Improving M&S Representation of the Soldier

Manuel Diego, Clayton W. Burford
US Army RDECOM Army Research Laboratory
Orlando, Florida
manuel.r.diego.civ@mail.mil, clayton.w.burford.civ@mail.mil

Joseph S. McDonnell, Ph.D., Bert Davis, Gary Smith
Dynamic Animation Systems, Inc.
Fairfax, Viginia
joe.mcdonnell@d-a-s.com, bdavis@d-a-s.com, gsmith@d-a-s.com

Derrick Franceschini
StackFrame, LLC
Sanford, Florida
derrick@stackframe.com

INTRODUCTION

Models and simulations have not kept pace with the Army's shift in focus from systems to the Soldier. There are no engineering drawings of a Soldier that can help with a functional or component decomposition. This is for good reason; since individuals are not interchangeable, they do not react to the environment the same way, and so models must account for individual variability in reacting to stimuli. The United States (US) Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED) Simulation & Training Technology Center (STTC) initiated the Distributed Soldier Representation (DSR) research project to investigate those factors that affect Soldier effectiveness, identify where there are gaps in modeling those factors in current Soldier representations, and offer a service oriented distributed Modeling & Simulation (M&S) environment able to assist in filling those gaps. While we were confident we could integrate existing models, we did not know what they were, or how complete the current decomposition is. Accordingly, we proceeded with the following assumptions:

- **Commonality:** There is some degree of commonality among humans that can be abstracted and modeled.
- **Decomposition:** This commonality would allow a reasonable decomposition of the Soldier into component models.
- **Discoverability:** Current thinking on this decomposition could be discovered through a literature search of current Soldier affect models. This would enable us to identify gaps.
- **Analyst Interest:** Analysts would be interested in data from models that more faithfully represent Soldier behavior.
- **Soldier as a Complex Person:** We could (eventually) provide a representation of the Soldier as a complex person.

Also included in this assumption is the belief that a service oriented architecture is desirable and achievable. It may turn out that the interactions among Soldier attributes are too tightly coupled to allow for a robust, complex representation. It may be possible, at best, to represent only a few areas in high resolution, and the complete, real-time, high resolution Soldier simulation is to be realized in the future. Designing and implementing DSR as a collection of services provides flexibility to satisfy this unknown aspect of the effort.

Soldier Partition

The initial investigation had no preset ideas of what Soldier attribute models would be available. Consequently, there was no predetermined Soldier decomposition in mind. One thing the decomposition was to do was capture the human aspects of the Soldier. During the literature review, similarities in models became apparent, at least to the DSR Team. Through this literature search, we identified 11 areas of interest as an initial decomposition of the Soldier: Cognition, Morale, Soldier Resilience, Physiology, and Psychology, Unit Cohesion, Stress, Unit as a Complex Adaptive System, Leadership, Decision Science, and the Effects of the Soldier as a Family Member. The details can be found in the ARL Tech Report ARL-TR-6985, “A Study in the Implementation of a Distributed

Soldier Representation." An open future task is to validate this decomposition, and in the process, investigate the feasibility of a Soldier decomposition standard.

DISTRIBUTED SOLDIER REPRESENTATION SERVER (DSR SERVER)

There is no shortage of distributed simulations that portray the Soldier as an entity. One Semi-Automated Forces (OneSAF), Infantry Warfighter Simulation (IWARS), Combined Arms Analysis Tool for the XXIst Century (COMBAT XXI), Joint Semi-Automated Forces (JSAF), etc., all provide some Soldier representation to differing degrees of resolution and fidelity. It is not our intention to create a new entity level model. Rather, our intention is to find those attributes with mature models that we can make available to existing simulations that may benefit from them.

In order to create a test bed for examining these factors, we created a prototype DSR Server. Lightweight and web based, we were able to look at the best way to make models of these attributes available to external clients, beyond the distributed simulations listed above. We want to provide the analyst the ability to use the DSR modules in a minimal configuration, eliminating any unnecessary simulation overhead.

The DSR Server has the following initial objectives:

- Gain insight into providing common model services via a flexible interface.
- Gain insights into different model runtime strategies, including federated runtime, DSR Server runtime, and free-standing runtime.
- Gain insights into different strategies of model composition.
- Gain insight into the feasibility of aggregating Soldier models at a central service.
- Gain insights into adequate protocols and wire formats for configuration, initialization and runtime of aggregated models.
- Attempt to drive design from the perspectives of a model developer and an analyst user.

Our initial implementation has focused on the first three bullets. Future research will focus on the rest.

DSR Server Architecture Goals

We want the DSR architecture to have the following characteristics:

- Service oriented
- Variable fidelity (accuracy)
- Variable resolution (precision)
- Tailorable to the models

Service Oriented

Service Oriented is often a goal in the M&S community, but not always fully realized because of the mismatch between service providers and service consumers. Because of the need to provide context in our simulations, services in an M&S Service-Oriented Architecture (SOA) implementation are often fully formed representations of a system. In our case, we first wanted to find representations of our system (the Soldier) that could provide a decomposition of attributes that would benefit from externalizing some effects. We realized that could be unnecessarily restrictive. There are times when it might make more sense to provide a capability as an internal module, or to meet the analytical need, to tightly couple modules within one application. This tension between providing some reasonable level of desired completeness, while NOT duplicating any of the numerous simulations that already exist, is still an area we are actively working on.

Fidelity And Resolution

This is an active area of research for us. It is easy to visualize variable accuracy and precision when thinking about terrain, for example. The concepts of geo-typical and geo-specific, as well as the idea of post spacing, make these quantifiable discussions. Human factors such as Resiliency and Soldier as a Family Member do not lend themselves to such neat definitions of fidelity and resolution. However, our goal is to provide a capability driven by analytical data needs, and this requires a consideration of both fidelity and resolution.

Tailorability
This is another area of active research. Early on, we decided not to restrict the DSR project to any one transport protocol or middleware, such as Distributed Interactive Simulation (DIS), High-Level Architecture (HLA), Test and training Enabling Architecture (TENA), etc. This approach provided exploratory freedom in design of services using modern commercial best practices. For now, we have implemented the DSR Server using a Websocket implementation while we continue to explore expanding to multiple transport protocols or middleware.

Websocket Implementation
WebSockets represent a long awaited evolution in client/server web technology. They allow a long-held single Transmission Control Protocol (TCP) socket connection to be established between the client and server which allows for bi-directional, full duplex, messages to be distributed with little overhead resulting in a very low latency connection. Websocket connections are established via an HTTP upgrade request thus bypassing many issues traditionally seen when routing across network boundaries.

The WebSocket API and the WebSocket protocol are agreed upon standards for low-latency communication between Internet clients and servers. Originally a browser technology, WebSockets are reaching far beyond just web browsers and are becoming a cross platform standard for realtime communication between client and server. Most browsers have native WebSocket support, and WebSocket library implementations are handled within many programming languages.

WebSockets were chosen for DSR Server since they represent a standard for bi-directional low-latency communication between servers and clients. The standards first approach means that as developers we can finally create functionality that works consistently across multiple platforms. Connection limitations are no longer a problem since WebSockets represent a single TCP socket connection. Cross domain communication has been a part of our design from the beginning and it is dealt with within the connection handshake, allowing services such as DSR Server to offer a massively scalable, low-latency platform that can be used by anyone.

Initial Implementation
Our goal in the initial implementation was to leverage existing Soldier models while also creating a low-friction platform to host new and future models.

Two of the eleven areas of interest, stress and human physiology, were deemed mature enough to support the initial DSR prototyping and demonstration effort. The Effects of Stress (EoS) module was developed to address stress, as it relates to small arms accuracy. We will address physiology in the continuing development section below. The initial DSR Server system was created using existing facilities and the products of previous experiments, including prior STTC research that provided necessary insights into the feasibility of a distributed Soldier representation (Gaughan, Metevier, Fogus, McDonnell, and Gallant, 2013).

DSR Server
The initial DSR Server was designed with the following dictums in mind:

1. Allow persistent client connection architecture
2. Provide a web-based interface to allow clients to view modeled Soldier information real-time
3. Provide a collection of high fidelity models as modules (or services) that can be requested by the web client

To facilitate communication between the DSR Server and clients, the DSR Server provides web services interfaces. For simplicity sake, we assume the following for all clients/models interacting with the DSR Server:

1. Time is independently managed between the DSR Server and clients. It is expected that the DSR Server and simulation are able to maintain advancing simulation time consistently with wallclock time. This will be changed later to allow for repeatability and is addressed later in the section on implementation challenges.
2. For consistency, the DSR Server model changes only when the client updates simulation state. (E.g., time is not advancing in DSR when it is not advancing in a client simulation and vice versa).
3. All information is encoded in UTF-8.

Persistent Client Connection Architecture

The persistent connection portion of the DSR Server is written in the SCALA programming language and leverages the Typesafe Activator and Play frameworks. The DSR Server implements a Websocket interface to clients that handles the following messages:

- Connection protocol (how to initiate the Websocket, server responds with Accepted or Denied message)
- Handshake message (expected once connection has been accepted, when received client is free to send messages
- PropertyList Request (returns properties or modules available to clients, response is a list)
- Entity/Individual Combatant(IC) Registration (client sends to DSR Server what ICs to be affected by DSR Server, based on PropertyList)
- SimulationState (provides DSR Server with state of simulation)
- StressUpdate (response to following messages with a stress value – EntityUpdate and DetonationEvent)
- EntityUpdate (client message sent when IC status has changed within the simulation. Factors include change in location or health)
- DetonationEvent (client message sent when detonation has occurred. DSR Server calls EoS module to calculate stress based on whether ICs are within range of detonation location)
- WeaponFire Event (client message sent when IC fires a weapon. Response message includes an updated stress and updated accuracy modifier based on that stress)

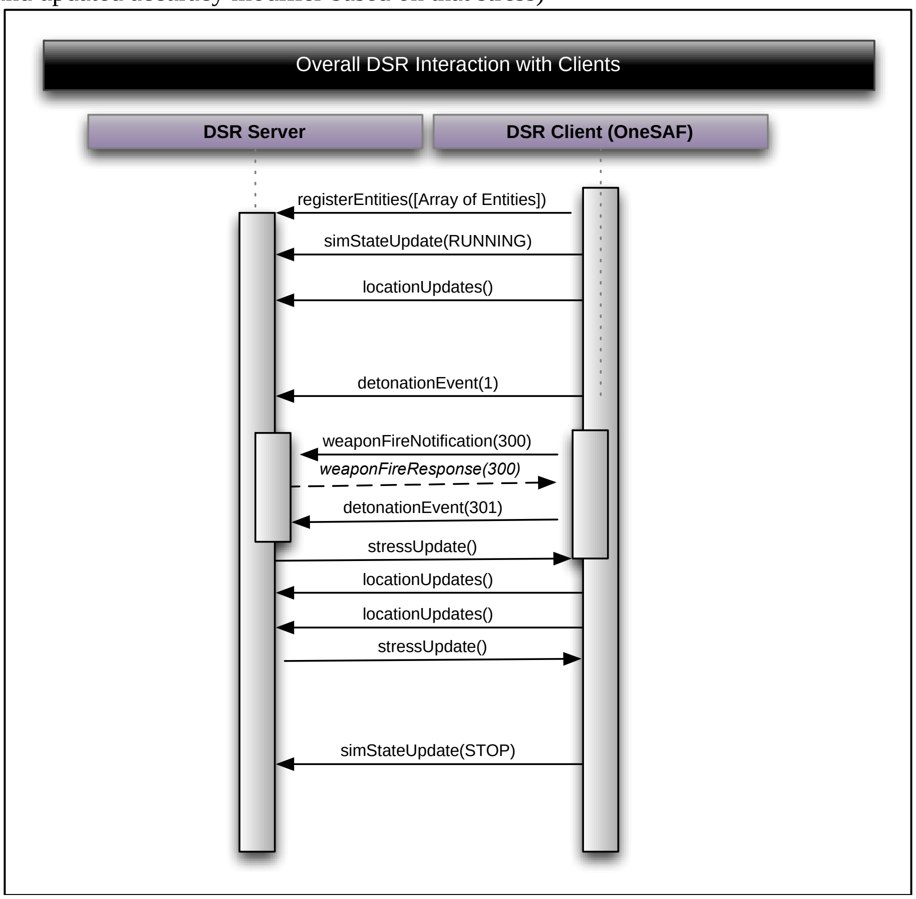

Figure 1. DSR Server interaction with clients

Web-based Interface

The web-based interface presented to the user is based on a mix of Hyper Text Markup Language (HTML) 5, Cascading Style Sheets (CSS), JavaScript and SCALA, in the form of scala.html files. The interface is designed to be flexible.

Modules

Modules encapsulate models within the DSR Server and then make those models available to clients. Modules register with the DSR Server during the server initialization process, and make the server aware of which specific messages they are set to handle. Clients send messages to the DSR Server, which calls the module that is set to handle that message. The DSR Server then gets the output of a module and then responds back to the client with updated information based on that module.

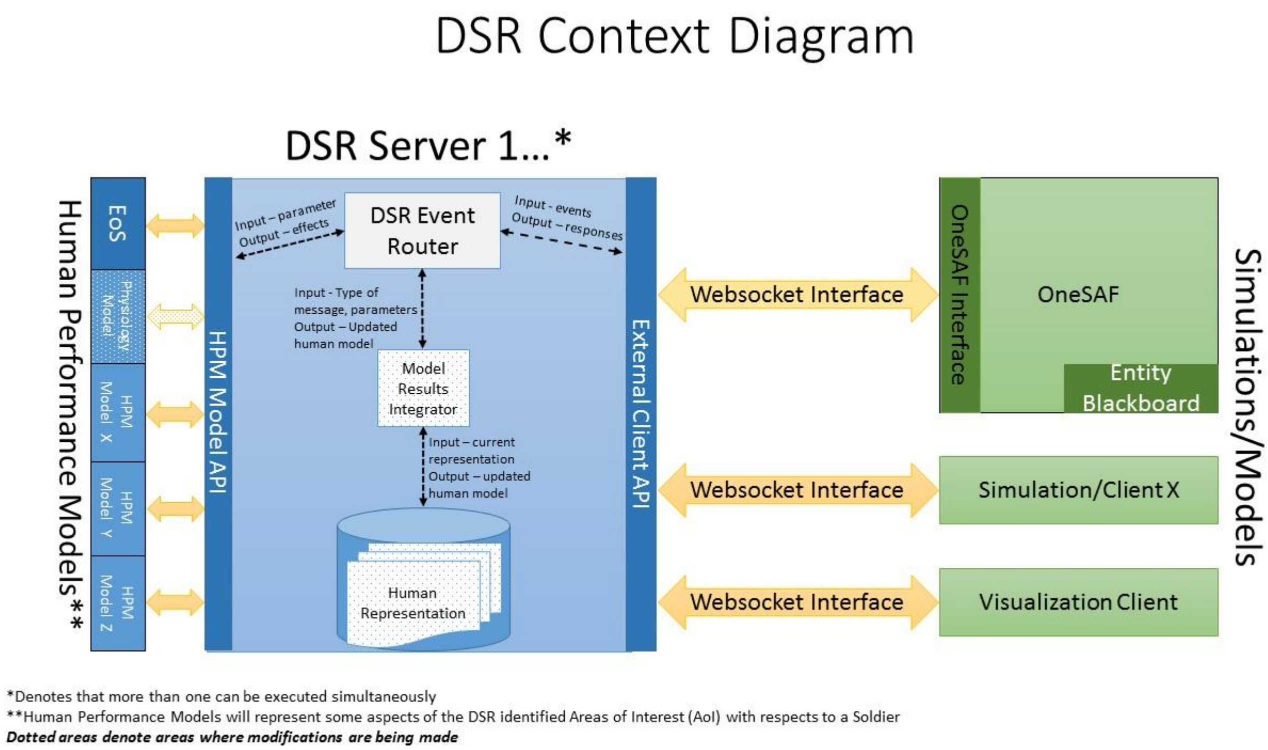

Figure 2. DSR System Overview

Effects Of Stress (EoS) Module

Based on our research, a Soldier's ability to effectively employ his/her personal weapon decreases when his/her stress increases (Sade, Bar-Eli, Bresler, and Tenenbaum, 1991). Thus as stress increases the probability of a Soldier hitting an enemy combatant decreases. While this seems to be well understood, we did not find any implementation of this concept. We created a "proof of concept" application for generating a representation of individual, dismounted Soldiers employing small arms (M9, M16, M4, M203, M249 and M2). The EoS module provides for decreasing levels of Soldier small arms accuracy with increasing levels of individual Soldier stress. OneSAF was chosen as the simulation to integrate with DSR Server. Stress levels provided by the EoS module impact the direct fire weapon accuracy, therefore using the effects to make the Soldier's weapon employment more inaccurate as his stress level increases. To visualize these effects, our team constructed a web-based tool as shown in Figure 3.

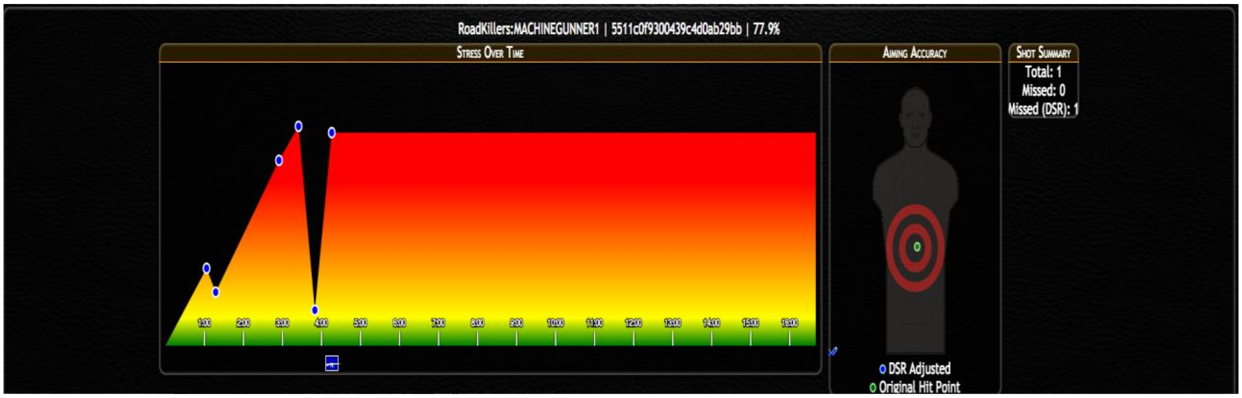

Figure 3. EoS Web Based Visualization

The time series data on the left of Figure 3 shows the cumulative stress of the IC. Note that the stress level can increase as the Soldier encounters more stressors, and decrease as the stressors dissipate. In this particular case, stress increased for this Soldier until just under ¼ of the way through the graph, at which time it dissipated. Just after it began to climb again, the Soldier engaged a target. The right hand side of Figure 3 shows a single weapon fire event from this Soldier while under stress. In this case, the Soldier was aiming for the torso of the target, and based on the default OneSAF algorithm, data, and scenario conditions, would have hit the target with the shot shown as a green circle. With the DSR adjustment for stress, the shot entirely missed the target to the lower left, as shown by the blue circle in Figure 3.

We realize that different individuals react to stress differently, and that some seem to have the capacity to handle more stress than others. For this reason, stress is provided as a percentage, and could be tailored to each individual. Additionally, it is possible to add a factor for chronic stress, so that the minimum possible stress level could be set to greater than zero. Table 1 contains the notional stress levels initially implemented in the EoS module. We are currently searching for validated data at the Army Research Laboratory Human Research and Engineering Directorate (ARL HRED) and the US Army Materiel Systems Analysis Activity (AMSAA), as well as soliciting input at conferences we attend.

IC Perception	Stress Factor
Not Under Fire	0
Perceived Fire	0.059
Perceived Near Miss	0.118
Peer Soldier Casualty	0.177
Immediate Leader Casualty	0.235
Self Wounded	0.294
Target Is Human	0.118
Minimum Possible Stress Level	0
Maximum Possible Stress Level	1

Table 1. EoS Notional Stress Levels

Implementation Challenges

When trying to get external clients to implement services from an external representation, the first challenge is always the cost of the first integration. Simulation components are usually developed to be as self-contained as possible, and the simulation will have to make internal changes to react and use the information provided by the service. Sometimes these changes can be trivial while others can be large development efforts. By selecting OneSAF we were able to leverage a simulation that we have years of experience with. The second challenge we have is scalability of the web service with regards to the simulation and the speed at which it needs inputs. Our initial application implemented a heartbeat within the simulation to ping the DSR Server for updated values. This caused a lot of latency with the messages that began to negatively impact the simulation. We have since updated the DSR server to eliminate the need for the client simulation to periodically request stress updates: stress updates are provided to clients minimally at a user specified time interval or when an entity stressor changes or the entity moves. A third challenge we have is supporting the analytical needs of the repeatability and causality. The initial architecture has independent management of time across all applications. This has to be modified which we are addressing in the next phase of development.

Implementation Benefits

An easily defined API and a JavaScript Object Notation (JSON) data-interchange format make for an easy implementation for sending and receiving messages. The client can parse the messages and take the information that they need for their application. This allows the flexibility to add information to the internal representation within the DSR Server and not affect the external simulation. If clients does not need the additional information, they ignore the data. Since the DSR Server didn't implement a simulation protocol there is no overhead associated with the federation of the service (and modules). The implementation also allows for a tailorable simulation that can answer specific questions. Quality of the simulation is based on data not on entity count.

LESSONS LEARNED

Incorporating New Clients

When incorporating new clients to leverage the DSR Server there is an implied need for modifications in those external clients. If there is no Websocket implementation within the client it will have to be added, along with the ability to handle the DSR Server outputs and use the data in an appropriate way. There may also be some slight modifications to the internal IC representation based on the client's representation of an IC. Based on our experience, these changes are less burdensome to incorporate for a client than adding an HLA interface for example.

Incorporating New Modules

When incorporating new modules into the DSR Server there is possibility that the DSR internal representation of the IC will have to be modified, which potentially affects the end clients. Based on scalability concerns, there may also be some changes or additional internal services necessary to support new modules. An example of that is the physiological module needs terrain data frequently. A terrain service, initially extracted from OneSAF, will be available to the DSR Server in order to alleviate the need to ping the client application at a high frequency for current terrain information for each IC being modeled by DSR. Using this terrain service allows perfectly correlated terrain services and data between DSR Server and OneSAF for our initial efforts. There will also be modifications of the DSR Server to accept and route new messages available, with updated documentation provided to client applications, and then those changes/additions will have to be addressed by the simulations if the client wishes to receive the new module data.

CONTINUED DEVELOPMENT

Physiology Module

As mentioned above, physiology was the second area of interest deemed mature enough to support the initial DSR prototyping and demonstration effort. We decided to leverage the Soldier Load Augmented Training Environment (SLATE) training application developed at STTC. SLATE runs on an Android based tablet (Figure 4) and is used to simulate a Fire Team-sized unit on foot, carrying typical combat loads. SLATE implements the United States Army Research Institute of Environmental Medicine (ARIEMS) metabolic expenditure equations to calculate the soldier's energy expenditure as the team traverses the simulated terrain. SLATE accounts for the following parameters: Terrain Grade; Soil Types; Elevation (Layered Terrain Format); Body Weight; Carried Weight (Soldier Load Equipment Distribution or SLED); and other factors supplied by SLATE. Exhaustion curves are based on FM 21-18 data trend lines for sustained work.

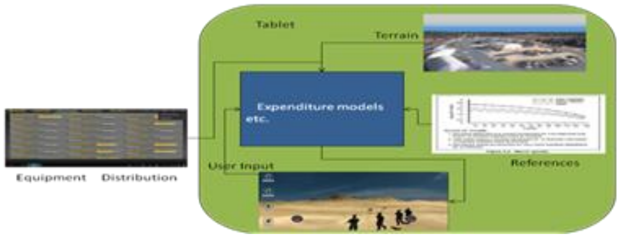

Figure 4. SLATE Tablet configuration

For the integration, we will externalize the physiology model as well as the route planning abilities. A terrain service will be made available to the DSR Server in support of this integration. The terrain service will be a standalone version of the OneSAF internal Environment Runtime Component (ERC). This service choice ensures perfect correlation between then physiology model terrain representation in DSR and those used by the internal mobility models in OneSAF. Standalone in this sense means that the terrain service will be a component that is separated from other OneSAF dependencies. For the purposes of this effort, dynamic terrain effects are not taken into account, although this could be added in the future. For this integration, updates from the physiology model in DSR will affect the OneSAF mobility model to affect a modeled soldier's fatigue level. With this integration we will accomplish:

1. Offloading the calculation of the individual's energy expenditure as it is affected by terrain to a reusable services. This is a computationally intensive activity. By directly integrating with a terrain service rather than making network requests for terrain information to a simulation, our approach minimizes latency for the DSR server to update mobility models.
2. Externalization of the route planning and energy expenditure functions, making these functions available to external simulations via the DSR Server.

The United States Military Academy (USMA) is also working a Capstone Project to take the area of interest Leadership and provide a model representation (Fletcher, McCabe, Truan and Walker, 2015). The next effort for DSR may be to integrate that module into the DSR Server as a new capability.

FUTURE RESEARCH

This current effort establishes a framework for integrating soldier behavior and effect modules into a Soldier server for use by combat models and simulations. Additional research is needed, and the following is a list of areas being considered by STTC:

- Research chronic stress and its effect on Soldier behavior pre-deployment, deployed and post-deployment
- Apply DSR to the Soldier Systems Engineering Architecture (SSEA) goals and soldier decomposition.
- Research mechanisms for dynamic resolution changes/modeling responsibilities. Have DSR adopt ownership of Soldier performance based on scenario criteria such as within a geospatial playbox (all soldiers in the urban environment) or within units that are currently in contact with red forces, etc. DSR would release responsibility for Soldiers not active or in a defined playbox.
- Investigate the use of new Java version 8 features including Parallel Streams to improve scalability of the DSR server
- Investigate the use of the new Java version 8 feature Nashorn to potentially enable DSR models to be written in scripting languages, such as JavaScript
- Investigate building a canonical JSON representation for use with SSEA, JavaScript and WebLVC
- Investigate support of area of interest processing for geographic filtering of DSR services
- Investigate architectural impacts based on implementing the Distributed Model Framework (DMF), a current development collaboration between STTC and USMA

CONCLUSION

As seen above, while much has been done, this effort is just in its initial stages. Other projects addressing Soldier architectural issues include NSRDEC's Soldier Systems Engineering Architecture (SSEA) STO and ARL HRED

STTC's Executable Architecture Systems Engineering Distributed Modeling Framework (EASE DMF). There appears to be a clear need for new architecture paradigms as well as the introduction of Soldier effect models. Some additional areas of active research that ARL HRED STTC is pursing to address this include the following:

Data
It is often said content is king. In our world, data is content. We are actively pursuing partners with data or models that represent some aspect of the Soldier, and how it affects his/her performance. We've come across a lot of data, but it is not necessarily in executable form. Validating the data is relevant to today's operational environment and that it can be used by simulations are two areas we must address if we are to provide analyzable results.

Soldier Decomposition
We need to validate, refine, or modify the Soldier decomposition. Our 11 areas of interest were the results of a bottom up assessment of the current state of Soldier modeling. We cannot attest to the completeness of this decomposition. As a result of the decomposition effort, the selection of Stress and Human Physiology were chosen due to the availability and maturity of models within their respective domains. We also have not evaluated the practicality of this with respect to an implementation within a service oriented architecture. Our initial attempt at this is on ongoing effort to integrate the Effects of Stress Federate with OneSAF to test its efficiency and to document the utility in the greater community.

DSR Architecture
We mentioned earlier that the language and development environment for the DSR Server needs to be reviewed and reevaluated. STTC, in conjunction with the United States Military Academy (USMA), is investigating a Distributed Model Framework (DMF) for assembling models using functional programming patterns (Kewley and McDonnell, 2014. Kewley, MacCalman, McDonnell and Hein, 2015). The high level goals include: easy discovery, understanding and integration of existing models; transparent manipulation of data so that the trajectory of state data can be seen from (exposed) input data to output analysis; promote efficiency in experiment design and execution; and be susceptible to parallel computation for large scale execution. We are conscious that whatever the final form of the DSR Server, it must not be so esoteric that the current user community is unwilling to adopt it.

REFERENCES

Fefferman, K., et al, (2015). Army Research Laboratory Technical Report A Study in the Implementation of a Distributed Soldier Representation, ARL-TR-6985

Fletcher, K., McCabe, S., Truan, T., & Walker, M., (2015). Distributed Soldier Representation: Implementing Cohesion in Combat Simulations. United States Military Academy.

Gaughan, C., Metevier, C.J., Fogus, M., McDonnell, J.S, & Gallant, S., (2013, October). *Executable Scenario Definition Using Datalog to Describe Simulation Capabilities,* NDIA Systems Engineering Conference.

Kewley, R., MacCalman, A., McDonnell, J.S., & Hein, C., (2015). *DEVS Distributed Parallel Architecture for Enterprise Simulation*, The Society for Modeling and Simulation International Spring Simulation Multi-Conference.

Kewley, R., & McDonnell, J.S., (2014). *Distributed Reactive Simulation,* The Society for Modeling and Simulation International Spring Simulation Multi-Conference.

McDonnell, J.S., Gaughan, C., Fogus, M., Burford, & C.W., Reed, D. (2014). *DSR Implementation Concept Overview,* Retrieved June 12, 2015 from www.dtic.mil/ndia/2014human/McDonnell.pdf

Sade, S., Bar-Eli, M., Bresler, S., & Tenenbaum, G. (1991). Anxiety, Self-Control, and Shooting Performance. *Perceptual and Motor Skills*, Volume 71, 3-6.