THE COMMON BATTLESPACE ARCHITECTURE – A DISTRIBUTED TACTICAL/ELECTRONIC/SYNTHETIC COMBAT ENVIRONMENT

Brett Vonsik
Lockheed Martin Corp.
Orlando, FL

Greg Mikkelson
Lockheed Martin Corp.
Orlando, FL

ABSTRACT

The modern battlefield upon which warriors engage in the deadly art of combat is an environment filled with increasingly sophisticated and complex weapon systems. Weapons, sensors, and transmitters utilizing new technologies and the entire electromagnetic spectrum have become commonplace and the knowledge required by warriors employing these systems has grown proportionally. Simulations of today's weapon systems and the battlefields in which they must operate have improved both in fidelity and richness, but their implementations have evolved in isolation from one another. Aircraft, ground vehicle, naval vessel, and dismounted infantry simulations have concentrated on the interactions between soldiers and their weapon subsystems, but this ownship-centric approach often results in battlefield abstractions, modeling only those subsets of the combat environment necessary to train specific tasks.

The current trend in manned simulation is to use vehicle mission computer programs, emitter and signature databases, and weapon systems intelligence in order to use the actual behaviors of the warrior's subsystems. This raises the bar of simulation fidelity and demands real-time data streams reflecting real world information. Simply put, current simulations of the modern battlefield are hard pressed to provide the fidelity required to drive today's manned simulators. Computer Generated Force (CGF) simulations, on the other hand, have focused on high-level battlefield abstractions, simply capturing the general essence of war and associated weapon systems. This approach may employ Monte Carlo methods, ignoring the detailed interactions between multiple weapon subsystems within a natural environment. Typically, CGF simulations have been and continue to be developed independent of manned simulator requirements.

Manned simulators and CGFs are now being asked to interoperate and accurately simulate complex interactions between intelligent subsystems, including sophisticated electronic combat. The limited bandwidth of affordable wide-area network technologies is a major roadblock to meeting this new challenge. Often, compromises are employed to integrate inconsistent data, algorithms, and differing levels of fidelity, making each implementation a customized solution despite DIS and HLA standards. Fair-fight, correlation, and consistency issues are commonplace due to isolated developments of manned simulators and CGFs without a common battlespace framework to work within.

This paper explores the increasing complexity of the modern battlefield and the resultant demands on manned simulators, CGFs, and their synthetic environment as the industry progresses into the realm of distributed simulation. This paper also discusses the limitations of current modeling approaches and proposes a distributed interoperable architecture designed to provide a consistent, correlated, expandable synthetic battlespace across distributed simulations.

AUTHOR BIOGRAPHIES

Brett Vonsik is a Project Director and staff systems engineer in the Advanced Distributed Simulation Technology II program at Lockheed Martin Information Systems in Orlando, Florida. Current and recent projects include modification of the AC-130H Crew Station Trainer, architecture development for AC-130U, Navy EA-6B, and Marine Corp Aviation electronic warfare systems and tactical environments, LMCore emissions architecture design, and the Special Operations Air Traffic Control program. Prior to 1997, Mr. Vonsik served as an MC-130E/H electronic warfare officer in the U.S. Air Force Special Operations Command at Hurlburt Field and Air Education and Training Command at Kirtland AFB. Mr. Vonsik has over 14 years experience in electronic warfare and 10 years experience in simulation technologies. He received his undergraduate degree from the U.S. Air Force Academy and a master in computer information technologies and resources from Webster University.

Email: brett.a.vonsik@lmco.com

Greg Mikkelson is a staff software engineer in the IR&D department for Lockheed Martin Information Systems, Orlando FL. His responsibilities include design and development for their LMCore simulation infrastructure and the integration of LMCore with legacy and future simulation systems. He has been principally responsible for the design and development of Emission Services in support of Electronic Warfare, the incorporation and expansion of LMCore's Tactical Entity Environment, and its transition from DIS to HLA. He previously served as the lead software engineer for the integration of the Abrams Common Software Library into the CCTT M1A2, test lead for CCTT manned module PPQT/FQT, and he held design and development responsibilities within the CCTT PVD and MCC CSCIs. In a life prior to simulation, he was proud to work for IBM, helping develop the Launch Processing System for NASA's Space Shuttle. He received his BS in Computer Science from North Carolina State University, Raleigh.

Email: Greg.Mikkelson@lmco.com

THE COMMON BATTLESPACE ARCHITECTURE – A DISTRIBUTED TACTICAL/ELECTRONIC/SYNTHETIC COMBAT ENVIRONMENT

Brett Vonsik
Lockheed Martin Corp.
Orlando, FL

Greg Mikkelson
Lockheed Martin Corp.
Orlando, FL

INTRODUCTION

The training and simulation industry has made great strides over the last decade in providing the Department of Defense (DoD) with technology capable of interacting and interoperating between multiple simulators for the purpose of conducting distributed training and mission rehearsals. Distributed Interactive Simulation (DIS) and now the maturing High Level Architecture (HLA) standards have established a common inter-simulation communications framework allowing independent nodes to interact and to interoperate at varying levels. Today, interoperability is a major concern to almost every DoD simulation program, new and legacy. As a result of the interoperability successes of the 1990's and a military in the midst of realigning itself to meet the defense needs of the 21st Century, an ever-increasing demand is being placed on the training and simulation communities to provide high fidelity simulations capable of modeling the real world in simulated synthetic environments to support both training and mission rehearsal. The use of actual weapon system avionics Operation Flight Programs (OFPs) and Emitter Identification Databases (EIDs) in manned simulators is growing and is at the forefront of new challenges as fidelity requirements are driven higher in all domains: manned simulators, Semi-Automated Forces (SAFs) or Computer Generated Forces (CGFs), and the natural environment that they must interoperate within. Simulators are now required to operate both independently in support of specific crew training needs and in concert with other manned simulators and SAFs to support all levels of training and rehearsal. The challenge facing the training and simulation industry is to create a common environment or " Battlespace" capable of modeling the natural environment (terrain and atmosphere), physical entity representations and interactions, weapons interactions, communications, emissions, and electronic warfare interactions with sufficient realism and consistency to ensure that the warrior's perception of the synthetic battlefield is nearly identical to that of the actual battlefield. Particularly important is the issue of "fair-fight" between interoperating participants within a simulation exercise. This is especially true in the domain of electronic warfare where emissions, signatures, and sensors interact in a complex interdependent manner that is not well understood by the community at large, resulting in simulation abstractions and implementations at differing levels of fidelity.

A Common Battlespace Architecture is required to provide any potential for a "fair-fight" among simulations and provide a realistic representation of modern warfare including the enormous volumes of communication, datalink, sensor, transmitter, and vehicle and countermeasure signature emissions comprising future battlefields and their simulations. This paper discusses current trends in simulation and the limitations of independent simulation architectures and modeling approaches. To address these issues and lay a foundation for future interoperability, this paper recommends a Common Battlespace Architecture for distributed simulation - a distributed interoperable architecture that is capable of providing a consistent, correlated, and expandable view of the synthetic battlefield environment to all participants within a Common Battlespace.

TRENDS IN TRAINING AND SIMULATION

The modern battlefield is a complex and daunting environment that presents imposing challenges to military crews, planners, and leadership. Precision-guided munitions, fire-and-forget self-guided weapons, cruise missiles, sub-orbital platforms equipped with a wide array of sensors, unmanned aerial vehicles (UAVs) loaded with sensors and weapons, real-time intelligence data feeds to command centers and in-vehicle systems, global positioning systems (GPS), encrypted and anti-jam communications and data links, lethal and mobile air defenses integrated with sophisticated command, control, communication, computer, and intelligence (C2, C3, C4I) systems, remote sensors spread across the battlefield, stealth technologies, rapidly reprogramable systems tailored to engage and defeat specific threats, information warfare systems, an enormous volume of electromagnet emissions, and the tried-and-true radio are all part of the modern warrior's reality; the warrior's adversaries are armed with similar systems, all with disruptive and lethal intent. The warrior's perceptions of the battlefield are extending beyond the individual, providing the opportunity for improved decision making. The combination of new technology and the modern warrior allows individuals and units to coordinate attacks with other forces, both similar and dissimilar, against hostile forces with chilling precision and lethality. While the complexity and capability of weapon systems has increased, so too have training requirements for soldiers, sailors, and airmen to ensure that they are able to take full advantage of the vehicles, sensors, weapons, and countermeasures at their disposal. Unit, force, and tactics training are becoming critically important to militaries around the world so that every bit of capability is squeezed out of costly and complex weapon systems. Computer-based training and simulation systems are an integral part of each warrior's military experience from the moment they enter specialty training throughout their entire careers.

Simulations of weapon systems have increased in complexity and capability in near lock step with their real world counterparts and are evolving towards an integrated view of the modern battlefield. Manned simulators are incorporating actual avionics and defensive system OFPs and EID software into their internal simulations. These sub-system simulations are capable of being rapidly reprogrammed the same as their real world counterparts, providing more accurate, current, and realistic training and mission rehearsals. Advanced sub-system simulations present few problems for isolated, internal equipment such as mission computers that receive limited and known sets of data within an expected range of values. However, simulations of offensive and defensive sub-systems such as fire control radars and electronic warfare systems demand realistic input data that are not readily available from arbitrarily abstracted simulations. Such data originate from the system's antenna after emissions pass through and are affected by the natural environment's atmosphere and terrain and by the objects moving through the natural environment: aircraft, ground vehicles, ships, and man-made structures. This levies an exacting demand on internally simulated sub-systems, natural environment simulations, and SAFs to feed realistic data to these OFP/EID driven systems. Abstractions of natural environments and SAF simulations need to conform to this new paradigm and evolve to the point that little or no difference is perceived between the real world and the synthetic world. While the DIS and HLA standards handle the mechanics of object interactions that occur at low data exchange rates such as passing entity position, orientation, and velocity, they fail to adequately address the high data volumes and rates required by these more complex, higher fidelity simulations. An interoperable architecture is still required that addresses these needs while minimizing or eliminating the differences between each simulation node's perception of the natural environment, objects, and emissions.

LEGACY SIMULATION ENVIRONMENTS

A modern manned simulator typically contains numerous sub-systems similar to that in Figure 1.

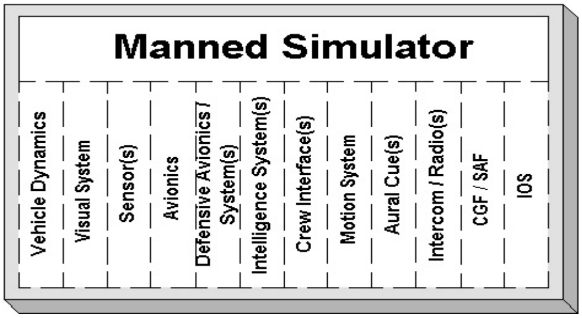

Figure 1. Stand-alone Manned Simulator

Historically, the weapon system dynamics and associated sub-systems, the natural environment, targets, and tactical threats have all been modeled internally so that crews could be trained to manage weapon system complexity, first to learn, then to maintain the skills necessary to effectively employ their weapon systems. The decision as to how these components are abstracted, simulated, and modeled continues to be at the discretion of individual simulator programs.

During the 1990's, many within the simulation and training community focused on networking manned simulators together in order to support the training of unit and combined force tactics. Initially, networking was made possible with closed and sometimes proprietary solutions allowing a small number of manned simulators to interact with one another in a limited manner. The exchange of basic vehicle-positioning data allowed manned simulators to perceive other manned simulators in their internally simulated sub-systems, but lacked the ability to interoperate in sophisticated ways where sensors, avionics, and tactical threats and targets were concerned. Recognizing these shortfalls, the DoD and the modeling, simulation, and training industry established the DIS and HLA standards as a means to simplify the networking of individual simulations and improve interoperability between them, but most of the effort was directed at basic vehicle, weapon, and communication interactions. Emissions and electronic warfare were abstracted at higher levels, leaving individual programs to interpret the standards as they saw fit and as requirements directed in their domain.

Meanwhile, command and control simulations supporting command staff personnel training in the management of human and equipment assets under combat conditions migrated from table-top, physical model simulations to computer-based exercises using computer-based SAF simulations to replicate combat forces operating under real-time conditions. The sophistication of SAF modeling grew throughout the 1990's. Manned simulator's internal CGFs improved in capabilities but retained their proprietary natures, uniquely modeling the natural environment, threat weapon systems, sensors, ordinance, behaviors, and tactics. Unique integration efforts specific to a particular manned simulator and CGF were required in order for both to be able to interoperate. Computer generated force simulations such as the Electronic Combat Simulation System (ECSS), Electronic Combat Environment (ECE), Effectiveness of Navy Electronic Warfare (ENEWS), Interactive Tactical Environment Management System (ITEMS), and Janus now provide adversarial air defense and, to a lesser degree, ground force entities for manned simulators. For these manned simulator dependent CGF systems, interoperability is enabled via gateways with other simulations, but their interoperability suffers limitations due to differing algorithmic and data modeling, conceptual abstractions of the battlefield, and interpretations of environmental interactions between synthetic and manned simulations. Each of these simulations followed their own unique modeling approach and abstracted view of the battlefield, providing certain strengths, weakness, and inconsistencies to the overall mix. Specialty simulations such as the Combat Electromagnetic Environment Simulator (CEESIM), Reconfigurable Infrared Simulation System (RISS), and Views also appeared and matured during 1990's. These systems continue to provide high fidelity electronic warfare simulations to equipment testers and certain manned simulators, but their overall capabilities fall short of simulating the total battlefield environment and threat behaviors required by the majority of general-purpose manned simulators today. A typical network of manned simulators, depicted in Figure 2, presents a daunting interoperability problem, especially if each node retains its legacy architecture and abstractions.

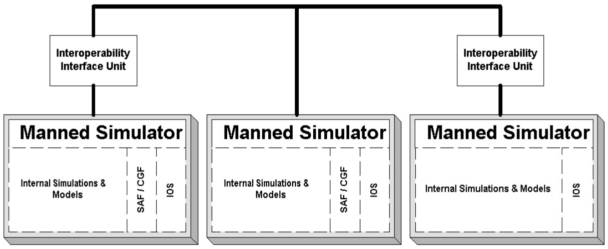

Figure 2. Typical Manned Simulator Network Topology

General-use, distributed SAF / CGFs that follow DIS and/or HLA standards emerged in the early 1990's. This marked the start of the transition away from traditional simulator architectures with internally integrated CGFs, towards a truly distributed simulation architecture. CCTT SAF, Modular Semi-Automated Forces (ModSAF), its derivatives Joint Semi-Automated Forces (JSAF) and ExportSAF, and its follow-on OneSAF Testbed, lead the pack for ground force SAF simulations today. Tactical Environment Network (TEN) and HLA Warrior (HLA variant of Janus) also round out this list of ground force SAF simulations. These simulations model air defense entities in addition to ground combat force entities, but do so with less fidelity and with reduced behavioral intelligence. This is due primarily to the development focus of these SAFs on ground combat rather than integrated air defense systems (IADS). In recent years, air defense SAFs have also been developed using DIS and/or HLA standards, such as Distributed Information Warfare Constructive Environment (DICE), Tactical Environment Simulation (TES), and The Next Generation Threat System (TNTS). However, these SAFs have concentrated their development on synthetic force modeling of the IADS with little attempt to model ground combat. These simulations suffer problems similar to their ground force focus SAF cousins, each with varying levels of abstractions and unique implementations of vehicle dynamics, sensors, ordinance, behaviors, tactics, natural environment, and emissions modeling.

Each SAF, regardless of its domain focus (ground, sea, air defense, or air forces), has strengths and weaknesses, but each remains unique in the way it implements entity, sensor, weapon, logic, natural environment, and emission models. Fidelity of synthetic weapon system models, their physical interaction with each other and manned simulator simulations, and their interactions with the natural environment remain varied. Manned simulators fair no better. Development and employment of military simulations span the spectrum from part-task procedural trainers, to part-task tactics trainers, to high fidelity weapon system trainers. Their internal models for vehicle dynamic, avionics, sensor, countermeasure, natural environment, and electromagnetic emissions remain unique due to system fidelity requirements, architectural approaches employed, and available technology. A less common network simulation architecture in Figure 3 retains all the problems of that shown in Figure 2, but adds additional complexity with the inclusion of an independent SAF. Again, vehicle dynamic, avionics, sensor, countermeasure, natural environment, and electromagnetic emission simulations remain unique. True interoperability is limited due to the differing levels of fidelity and abstraction in the models implemented in each independent simulation, both manned and synthetic, with sophisticated interoperability only becoming more illusive.

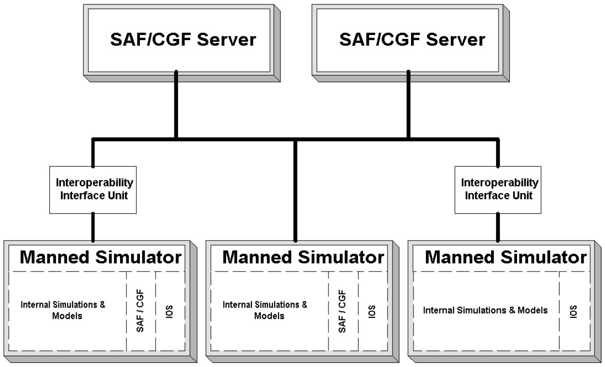

Figure 3. Typical Manned Simulator Network Topology with Distributed SAFs

As battlefield simulations continue to mature, incorporating increasingly complex interactions and higher fidelity models, the redundant development of internal models and CGF capabilities becomes exceedingly expensive and divergent – it is no longer cost effective to develop similar, increasingly complex functions by various simulation houses with differing perspectives, biases, and requirements. The simulation community is aware of the advantages, even the necessity, of distributed interoperability, as evidenced by the growth of DIS based simulations over the last decade, and the drive by all branches of the U.S. military to develop HLA compliant solutions. By compartmentalizing subsystems, each component can be developed and groomed by focused, domain specific experts. With such an approach, each component more accurately emulates its real world counterpart and ensures a consistent presentation to each participating sub-system. This approach also reduces development costs for each platform by sharing the cost of common components across multiple users. With proper planning and design, a common approach can be developed that provides for the integration of both new and legacy simulators and models within a distributed network. Architected properly, this common approach can provide for the sharing of resources and functions across participating platforms, and allows exercise/scenarios to scale up from stand-alone, to team/unit/flight, brigade/squadron, and even joint operations.

INTEROPERABILITY PITFALLS

Inconsistent Data and Algorithms, Variable Fidelity, and Unfair Fight

The road to interoperability is not without challenges. Independently developed simulations can be expected to incorporate differing, even competing algorithms and philosophies to common functions. As they are made to interoperate, differences in architecture, algorithms, fidelity, and philosophy will invariably affect their interaction and simulated interplay. For example, different simulations may incorporate unique algorithms for calculating inter-visibility (line of sight), radar cross section, signal strength and attenuation, ballistic fly-out, susceptibility to counter-measures, to name a few; some may outright omit certain calculations, simulating only the end result and/or assuming optimal, average, Monte Carlo, or minimal conditions and results. Slight variances in any single calculation may result in unfair advantages for one participant over another and/or unrealistic outcomes from the resultant interactions. For instance, a high fidelity entity may utilize "known" standoff distances that are ignored by a lower fidelity threat. The low fidelity threat may then engage the target from an unexpected, unrealistic distance if the low fidelity threat simply applies Monte-Carlo methods to determine the results of the engagement. In such a scenario, any evasive maneuvers or countermeasures employed by the target may not have an affect on the final outcome; neither of the crews would accurately practice or learn the benefits of properly applied procedures and the resultant effects of employed counter-countermeasures. While major discrepancies may be readily detectable and corrected, at a price, other differences may be more subtle and nefarious, resulting in false expectations and negative training - at an even greater price.

A further example of this problem can be illustrated in the basic interactions between a targeting air defense radar site and an attack aircraft as depicted in Figure 4. Two simulation nodes, a manned aircraft simulator and a Surface-to-Air Missile (SAM) system generated from a SAF, are attempting to interoperate beyond basic spatial interactions. Their positions (x, y, z) and orientations (pitch, roll, heading) are communicated over the network using DIS or HLA protocols using a Geocentric Cartesian coordinate (GCC) reference coordinate system. The natural environment, electromagnetic propagation, and sensor emissions data is modeled uniquely and differently in each simulation. The aircraft simulator has on-board defensive avionics equipment, a radar-warning receiver (RWR) capable of detecting the radar emissions from the SAM, an expendables countermeasure sub-system (chaff), and a self-protection jammer. The RWR and jammer are advanced simulations sporting actual system OFPs and EIDs and requiring real world data to identify and determine countermeasure employment. The aircraft simulator has a Digital Radar Landmass Simulation (DRLMS) to simulate the aircraft's on-board ground attack radar and models the terrain in successively coarser resolution planes as the range from the aircraft simulator's location increases starting with 3 arc seconds (approx. 300 feet) spacing between elevation posts and growing to 30 arc seconds at the extent of the DRLMS terrain database of 400 nautical miles (nm). Additionally, the aircraft simulator models the atmosphere in 20 altitude layers with each layer modeling temperature, pressure, precipitation, winds, and clouds independently. The SAF simulation has a generic SAM logic model to determine changes in weapon system mode state (off, search, track, fire), arbitrary mode transition conditions (pre-determined fixed ranges that trigger the SAM's mode state change when an entity enters them), and a basic emission parametric set (frequency, transmit power, and pulse repetition frequency) taken from unclassified sources; models the atmosphere in a single monolithic state; does not model atmospheric propagation of radar emissions; and models terrain at 3 arc seconds over a 60 nm by 60 nm terrain patch contained within the overall gaming area of the aircraft simulator.

The engagement starts when the aircraft position, transmitted over the network, updates the aircraft position and breaks the SAF SAM's pre-defined range condition set to change the SAM's mode state from “off” to “Search”. For the purposes of this example, the SAM's transition range is 90 nm and does not use aircraft radar cross-section (RCS) based on radar transmission power, atmospheric propagation, and reflected radar energy from the aircraft based on aspect angle to determine aircraft detectability. The SAF determines that the SAM has a direct line of sight to the target aircraft. The SAF transmits the SAM's entity state and emissions data (frequency, power, and PRF) to the network. The aircraft simulator receives the entity and emissions update and interprets them using internal algorithms and models. The aircraft simulation determines that a mountain stands between the aircraft and the SAM (the SAF's terrain patch does not extend far enough to see the mountain) and disregards the emissions information due to the SAM being occulted by the mountain. The manned simulator's aircrew are unaware of the computer interactions at this point and consider themselves safe because of their mission planning and route selection to keep the mountain between them and the known SAM site. The SAF continues to send entity state and emissions updates on the network and the aircraft simulator must continue to expend computational resources to determine that the SAM is occulted and should be ignored. The aircraft range to the SAM decreases to 80 nm, causing the SAM to change to a tracking mode. The SAF transmits new emissions updates indicating the SAM's state change. The aircraft simulation determines that a mountain still stands between itself and the SAM and continues to ignore it. The range between the aircraft and SAM decreases to 75 nm causing the SAM to change modes once again, this time to a missile firing state and spawns a new entity . . . a missile speeding towards the aircraft. The aircraft simulation continues to determine that a mountain exists between the SAM site, the missile, and itself and continues to ignore both. When the range between aircraft and SAM site decreases to 70 nm the missile closes to within 5 nm of the aircraft. The aircraft simulator determines that the missile is within a direct line of sight after having passed through a mountain (the missile is no longer occulted), references the SAF's SAM emission update for parametric data, and provides the data to the ownship RWR and jammer. The SAF's basic emissions data is out of the anticipated value range specified in the sub-system's classified database or EID, resulting in the RWR presentingan incorrect audio tone and symbol on the RWR display. The RWR and jammer OFPs and EIDs need real world data to determine emission identification and do not have enough information to correctly identify the emission, resulting in an "unknown" symbol being placed on the RWR display and no jamming initiated. Additionally, the aircraft simulator has misinterpreted the transmit power of the SAF's emissions as power received at the aircraft and passes this to the RWR without performing an attenuation calculation. This is due to the positive occult indication it continues to get from its internal calculation for the SAM radar. The manned simulator aircrew are momentarily alarmed by the RWR indication due to the high volume of the audio tone, but calm down when they see it is an "unknown" signal; likely a search radar not operating in a normal or anticipated mode or frequency. Moments later the crew is shocked by the sudden appearance of a missile emerging from a mountain side in the out-the-window visual displays approaching their aircraft. They only have time to react defensively using a breaking maneuver, no chaff is expended, and the jammer sits idle. The aircrew visually determined the missile missed the aircraft by more than 200 feet as it passed. The SAF determines that the missile passed close enough to the aircraft (100 feet) to warrant a proximity detonation of the warhead (latency over the network was not able to accurately account for the last ditch maneuver by the aircraft). The SAF issues a detonation event over the network with the manned simulator aircraft as the target of the detonation. The manned simulator receives the detonation update and calculates damage against the ownship aircraft, killing it and giving the aircrew a "green screen of death". The manned simulator aircraft is removed from the network exercise and its aircrew is left to wonder what happened.

In this example, a comedy of errors occurred that are not atypical of today's distributed simulation environment. First, the SAF's low fidelity sensor model and logic allowed the SAM to engage the aircraft at unrealistic ranges. The aircrew would have been safe under real world conditions due to correctly performed mission planning, having placed the mountain between their ingress route and the SAM site and armed with the knowledge that their aircraft's RCS would prevent detection by the SAM's radar beyond 40 nm. Second, the aircraft RWR and jammer models received an emissions update with too little data to correctly identify and counter the SAM's radar. Had the mountain not occulted the SAM radar, the aircraft RWR and jammer models still may not have detected the radar signal due to the calculation from the manned simulator's emissions propagation model attenuating the power of the SAM's radar emission below the threshold required by the ownship's defensive systems to detect and consider the SAM emission a concern. Had the aircraft detected and jammed the SAM, a simple SAM mode change (Fire to Search) would have been the result, assuming that the SAF allows for its threats to respond to electromagnetic counter-measures. In reality, the SAM would have been capable of maintaining track while being jammed as the aircraft jammer would have jammed the missile guidance signal and not the target tracking radar, sending the missile into an erratic flight path away from the aircraft. Third, the missile flew through the mountain in the manned simulator's visual system, reducing the aircrew's confidence in the simulator. Finally, the aircraft was hit and killed by the SAF's missile despite the aircrew's visual confirmation that the missile flew wide. Again, unrealistic and the aircrew's confidence in the simulator is reduced at best and destroyed at worst.

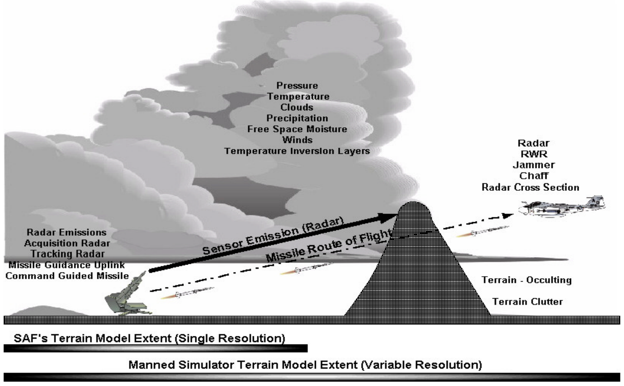

Figure 4. Simple Interoperability Example

True interoperability requires that all participants be presented with a single, consistent, realistic view of the common battlespace and the interactions therein. While it is largely incumbent upon the individual simulation components to ensure realistic simulations of their real world counterparts, it is the collaborative responsibility of the distributed system as a whole to ensure commonalty and consistency across those components. While the individual behaviors and models of each simulator may be architected and developed independently, a singular view of the common battlespace must be ensured by a common philosophy and should be governed by a common set of data, algorithms and services.

LAN, WAN, Bandwidth Latency

In traditional stand-alone simulators, manned or otherwise, the entire battlespace (players and environment alike) is contained within the local processor and all known parameters are immediately available to all portions of the simulation. As a result, updates and interaction rates were limited only by the processing power and internal memory/bus capabilities of the hardware platform within which they ran. However, the distribution of functions across a network imposes latency and bandwidth limitations upon the simulation. Today's 100 Mbit Ethernet LAN technologies allows for peak, sustained network data traffic in the 20 Mbit/sec range and Gigabit Ethernet has the potential to increase sustained data bandwidth capacities into the hundreds of Mbits/sec range. An exercise consisting of a few dozen entities, all capable of simultaneously passing numerous electronic emissions between electronic sensor, surveillance, countermeasure, weapon, and communication systems, presents a bandwidth challenge to affordable LAN architectures. Despite the capabilities of the latest network technologies, WAN bandwidths are typically much more restrictive, often as low as 1.5 Mbit/sec for a T-1 connection, due to the reality of cost. Bandwidth requirements to support tactical and electronic combat interactions described above will not be possible over affordable WAN leased lines if all associated parametric data needed by simulations using actual OFPs and EIDs is transmitted over the network on a pulse-by-pulse or near pulse-by-pulse basis. This is especially true when voice and digital data communications are added to the overall bandwidth demand. Minimizing latency of time-based activities such as physical proximity of multiple entities, communications, emissions, weapons effects, and countermeasures is critical in achieving the perception of a common battlespace for all participants. Many high-speed subsystem to subsystem interactions occur too fast (e.g. receiver/jammer/countermeasure thresholds are measured in increments of milliseconds) and require data volumes too taxing for existing network technologies. Rather than obtaining information about remote tactical/electronic participants directly from the network, consumers of such data should employ a local proxy that provides local regeneration (IEEE Std 1278.1) of the data. The proxy has the responsibility of maintaining entity state for the consumer, and where network latency or bandwidth issues preclude instantaneous updates of entity state from the master entity, the proxy must dead reckon the physical state of entities and emissions based upon common databases and parameters provided by the remote master. While the master is still responsible for determining the behavioral and physical state of the modeled component, the master need only distribute those data necessary for the remote proxies to determine the physical state using common agreed upon local regeneration algorithms and associated databases. Local regeneration of tactical, emissions, and signature data is necessary to minimize network traffic in LAN configurations, and essential for WAN operations.

Myth of the DIS/HLA Panacea

Obviously, it is not enough to simply connect a wire between two simulations and expect them to interoperate – certain protocols must be agreed upon and adhered to. However, even a common protocol is not sufficient to ensure interoperability. While a protocol defines which data will be shared and formats used, it does not define how that data will be used or transformed, nor does it ensure that the data will be interpreted or processed in a consistent manner across all players – in short, a protocol defines common data, but does not provide the common philosophy and services necessary to ensure a common, consistent, singular view of that data and the resultant battlespace. While the DIS community has evolved certain basic philosophies and tenets governing distributed simulation, HLA effectively portions the distributed community into separate federations, each of which is free to define their own data formats and philosophies. While a given HLA federation has greater flexibility to define and control its own interfaces, the resultant diversity only complicates future efforts to interface between such federations, much less to ensure that they are truly interoperable.

The problems identified previously are all too common in today's simulations when attempting to make them interoperate in an exercise. The issue is "fair-fight" and the issue is growing larger as manned simulators and SAF's become more sophisticated. Manned simulators and SAFs continue to model the physical world and weapon systems "their way" and lack the common ground to effectively interoperate. The major problem areas affecting the "fair-fight" are:

1) Variable modeling fidelities
    a) Agreed upon intelligence data and algorithms supporting the modeling of sensor emissions and scan characteristics, vehicle and expendables signatures, weapon systems, defensive avionics systems, and the critically interdependent interactions between these systems
    b) Weapons modeling (guidance, fusing, warhead types, and warhead yields)
    c) Tactical and behavioral modeling of synthetic forces for both single vehicle and combined force operations
    d) Atmospheric and natural environment modeling
    e) Database modeling and correlation between database (visual, terrain)
2) Inconsistent data and algorithms
    a) Lack of consistent entity signatures modeling throughout the electromagnetic spectrum (visible light, contrast/ultra-violet, infrared, radar, radio, & acoustic)
    b) Lack of consistent emissions parametric data and scan characteristics modeling (visible light, contrast, ultra-violet, infrared, radar, acoustic, communications, power, & seismic)
    c) Algorithmic calculations involving the interactions of emissions, signatures, atmospheric models, terrain occulting and radar and infrared clutter models, sensor models, and vehicle dynamics
3) LAN, WAN Bandwidth limits
4) LAN, WAN Latency

COMMON BATTLESPACE ARCHITECTURE

What it is

While individual federations of interoperable components will certainly be developed to meet current requirements, each is likely to contain it's own unique and proprietary solutions if they do not agree upon a common approach in advance. Without a common understanding, these progressively complex simulations will continue to diverge; each federation will become an island onto itself, resulting in a collection of stand-alone federations. Without an over-arching plan, the inter-federation interoperability required for tomorrow's joint operations will suffer difficulties akin to those faced by today's stand-alone simulators, but on a grander scale. To address these issues and to lay a foundation for future growth and interoperability, the simulation and training community must develop a road map to tomorrow's solution, a Common Battlespace Architecture. As discussed previously, the crux of the matter is to ensure a fair fight through common data, algorithms, and models; any solution must ensure that all participants perceive a complete, consistent, common view of the synthetic natural/tactical/electronic warfare environment (battlespace). This solution must be able to support current requirements while allowing the flexibility to address domain specific requirements as well as the growing complexity and interdependencies of the modern battlefield.

Since the Common Battlespace Architecture describes a distributed network of interoperable simulators, it actually contains two separate, but interdependent architectures – that of individual simulators (node architecture), and that of the network aggregation (network architecture). The Common Battlespace Architecture is, in effect, an architecture of architectures. The proposed network architecture is conceptually similar to those evolving out of DIS and HLA – a network of manned and SAF simulators wherein each node is responsible for modeling and controlling its local entity(s), communicating the states and interactions for each of its entities across a common network, and interacting with those entities controlled and communicated by the other nodes, as depicted in figure 5. Each manned simulator models a single vehicle which interacts with vehicles/objects generated by other manned and SAF nodes, while each SAF is a public service, modeling support, target, and opposing vehicles which interact with all other participants on the network. While this recommendation may be old news to the distributed simulation community, it is a new paradigm for many in the stand-alone community who may envision a different topology based on interconnected "all-in-one" simulators, each with their own private CGF/SAF. In addition to manned and SAF nodes, the networked architecture can contain a variable number of Battlespace Servers capable of controlling various battlespace attributes and/or off-loading processing from any of the distributed Battlespace Services.

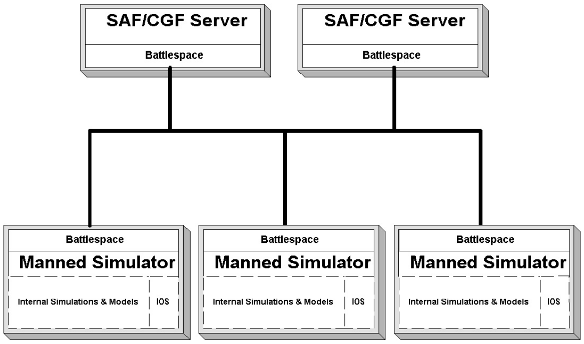

Figure 5. Common Battlespace Network Architecture

In order to ensure interoperability through commonality, each node within the network should adhere to the architecture depicted in Figure 6. This architecture places common services between the application (individual simulation) and the network (and thus the remote entities). By controlling which data is shared across the network, and how that data is interpreted and presented to the application, the Battlespace Services ensure that a single, consistent, common view of the Battlespace is presented to each application. While the Common Battlespace Architecture encapsulates each local simulation(s) into an application apart from the network and Battlespace Services, it imposes no internal architecture on the application itself. Simulations interface to the network through the common architecture services while retaining freedom and flexibility in implementing their own internal architectures. This not only provides the freedom necessary for each new model to be tailored to its domain specific requirements, but also allows legacy simulators with various architectures to be incorporated into the interoperable framework by modifying them to utilize the same Battlespace Services.

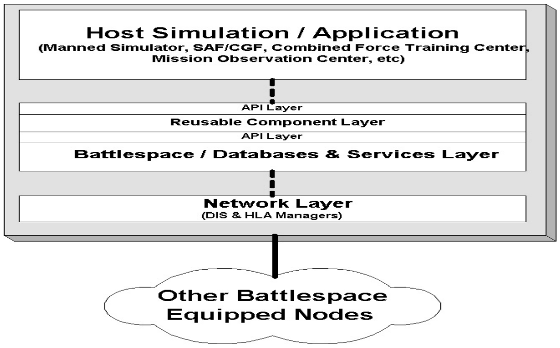

Figure 6. Common Battlespace Node Architecture

Conceptually, a simulation represents one or more real world entities operating within a real world battlespace. Within the proposed Common Battlespace Architecture, therefore, each vehicle or entity is viewed as a self-contained object modeled by a domain specific Application, while the physical space in which they operate and interact is managed by a set of distributed Battlespace Services. By providing a local proxy representation of each physical attribute and component within the battlespace, the Common Battlespace Architecture ensures that a single, common, consistent view of the battlespace is presented to each participant (Application), while leaving the Application unaware, and independent of the details of the underlying network, associated protocols, models and algorithms. This architecture allows Application specific simulations to concentrate on crew cueing/systems operations (in manned simulators), platform/aggregate force behaviors (in SAFs), and vehicle dynamics (in both). Each Application interfaces with the Common Battlespace through a standard Application Programming Interface (API), providing those data required by the Services to accurately represent the entity to others, and requesting those data required by the Application to model its performance within the Common Battlespace. The Services, in turn, manage and present the fair-fight physics and state of the battlespace, transmitting across the network only those tactical, emission, and natural environment data which are essential for the local regeneration by the remote proxies (Battlespace Services) in other nodes. The Services can be replaced or updated over time, evolving to meet the expanding needs of the training and simulation community; however, as long as participants use the same, or compatible, versions of the services, they will continue to perceive a singular view of the battlespace.

The Battlespace Services themselves are a collection of distributed proxy servers for maintaining and distributing the states of all objects, interactions, and attributes within the common battlespace. Much as the Application architecture is not restricted by the Common Battlespace Architecture, the details of the Battlespace Services remain conceptual within the architecture. However, since they must simulate the entire battlespace to each participant, the Services are expected to parallel the real world tactical, physical, and natural environments of interest, as well as to provide a controlled access to the network itself. The Tactical Environment must manage entity states, warfare interactions, logistics, emissions, ordnance, and entity signatures consumed by each Application and the services themselves. The Physical Environment must provide common physics based models, as well as typical "Mission Functions" required to provide occulting, Line of Sight (LOS), Height Above Terrain (HAT), radar and infrared surface clutter, collision detection, and geometry intersect determination of entities within the Tactical Environment of the Tactical Environment. The Natural Environment must manage the local instance of the Synthetic Battlespace's terrain, atmospheric, and relocatable model database used by Mission Functions and the Tactical Environment.

Tactical Environment:

Entity State – Manages entity state parameters to include position, velocity, orientation, configuration, sensor mode state, weapons state, and active behavior logic commands.

Warfare – Warfare manages and determines weapons fire and detonate events for consumption by the behavioral application and reports the same to remote Common Battlespace systems for updates to local Synthetic Battlespace databases.

Logistics – Logistics manages and determines logistics related data of entities for consumption by the behavioral application and reports the same to remote Common Battlespace systems for updates to local Synthetic Battlespace databases.

Emission Services – Emission Services provides for all electromagnetic emissions (from power and sound to cosmic rays) in the environment. Emission Services must simulate emission characteristics and scan patterns as defined in a common Emissions Database and specified by the controlling models in the reusable components and application layers. Emission Services must locally regenerate emissions and provide the simulated data to sensors, seekers, and system models in the reusable component and/or application layers. Emission Services must provide to sensor and seeker models: entity/target detection and tracking data, measurement and signal threshold testing (filtering) of emissions propagated through the Synthetic Battlespace, calculated background clutter caused by terrain and surface waves, impact of atmosphere, weather, precipitation, clouds and the sun on emissions, effects from chaff, flares, and other expendables. Emission Services must also maintain the Emissions Database for use by the manned behavioral and SAF Applications and other Battlespace Services.

Ordnance Services – Ordnance Services manages the Ordnance Database and calculates the "flyout" of all ballistic objects (e.g. missiles, bullets or rounds, rockets, bombs, and aerial mines). The basic physics required to aerodynamically "fly" objects is the same regardless of the ordnance type. Weapons with propulsion use the base ballistic flyout physics combined with thrust vector data. Guided ordnance, powered or not, also receive target/vectoring guidance commands from either the Application or reusable component models.

Signature Services – Manages electromagnetic signatures for individual entities. Signatures (e.g. radar, IR, visual/contrast, UV, and acoustics) must be provided at predetermined and agreed upon levels of fidelity in a sphere around individual entities -- 360 degrees horizontal by 360 degrees vertical. Fidelity of signatures can be tailored for every one, two, three, . . .10, . . . 20, degrees etc as required for an entity throughout the entire electromagnetic spectrum if needed. Signatures for entities can be stored for individual configuration loadouts, engine(s) state, etc. Expendables signatures, such as chaff and flares, will be based on expendable type and quantity. The signature for a given entity will be identical in each node's battlespace, ensuring consistent data modeling for manned and SAF Applications interfacing to the Common Battlespace Services.

Physical Environment:

Standard Physics Modeling – To ensure a common perception/representation of the physical world in which we operation, common algorithms must be identified and provided to all participants via Common Battlespace Services. Examples include 3D spatial calculations, ballistic flyout calculations, atmospheric and terrain effects on emissions, coordinate conversions, and translations between standard measurement units.

Occulting and Line-of-Site (LOS) – Occulting services must perform LOS calculations on a uniform level-of-detail (LOD) terrain database within the Common Battlespace environment. The terrain database size and LOD can be tailored based on user needs and computational resources, but occulting results will be consistent throughout the gaming area regardless of manned simulator or SAF entity locations. Occulting services must provide data such as background radar, infrared, and visual clutter calculations and path attenuation calculations for all manned and SAF sensors in the scenario eliminating this burden on DRLMS in manned simulators.

Collision Detection/Determination – Collision detection of entities tracked in the Common Battlespace environment must be determined and provided to the reusable component and application layers for damage assessment.

Natural Environment:

Terrain – Manages and maintains a copy of the gaming terrain database for use by the Application and local services. Terrain database fidelity or LOD can be tailored based on need and available resources. Terrain database formats such at CTDB7 should be considered for this database due to its wide use by simulations presently. Global Coordinate System (GCS) or GCC reference frames should be implemented to support both surface and aviation simulations that demand large databases (more than one geocell).

Atmosphere – Manages and maintains a copy of the gaming atmospheric database for use by local services. Atmospheric database fidelity or LOD can be tailored based on need and available resources and will be updated by an Environmental Data Server periodically as necessary.

Relocatable Objects – Manages and maintain a copy of the gaming relocatable objects database for use by local services and interactions based on weapons targeting. Relocatable objects moved, destroyed, or altered by any simulation within the exercise must be reflected in each local Common Battlespace environment.

How it works

The Common Battlespace Services provide a common simulation interface between all applications within a distributed simulation. Fair-fight issues are largely eliminated in this architecture by the use of identical data structures, tactical and natural data, emissions and signature data, algorithms, and processes between all participants. With local regeneration of intensive tactical and electronic combat, network traffic is reduced to only those fundamental data and parameters necessary to drive each proxy model, rather than continuous updates (e.g. 60Hz state changes, or worse); local regeneration significantly reduces latency and demands on LAN and WAN bandwidth. Correlation of tactical and natural environments and event consistency is achieved through the Common Battlespace. Furthermore, load balancing of entity data processing can be optimized over the network topologies through HLA ownership transfer when processing overruns occur within any single node. A standard API to a common set of services not only provides and supports consistency across Applications, but also encourages the development of reusable onboard components. By adhering to a common standard architecture, each federation of simulators resolves many of its internal fair-fight/interoperability issues while positioning itself to utilize services and components developed by other components adherent to the architecture and to interoperate with other federations in ever larger, joint operations. Adherents to a Common Battlespace Architecture, in effect, create a federation of compatible federations.

Since the Common Battlespace Services represent the entire tactical, physical, and natural environment to all participating applications (entities), the services can be viewed as a Common Synthetic Environment. This Common Synthetic Environment serves as the single, Common Battlespace repository for all data concerning entity(s) geometry, collision detection, emissions, signatures, atmospheric conditions, weapons detonations, and necessary deadreckoning. The Common Battlespace provides SAFs with the same common network interface, common simulation services, and common simulation component libraries that exist on the manned simulators. The SAFs in turn provide the behaviors of all platform and aggregate forces that it manages, much as manned simulators (and their crew) provide the behaviors of manned platform and aggregate forces. Each application processes its ownship and guided weapon geometry and provides that geometry to the Common Battlespace for consumption by other simulations. Weapons not requiring an active feed-back loop with crew operators in manned simulators (bullets, rockets, fire-and-forget missiles, laser-guided missiles, laser-guided bombs, GPS guided bombs, dumb bombs, etc) are provided to the Common Synthetic Environment for geometry calculation by the Ordnance Services or transferred to a Common Battlespace Server if local computational resources are limited. Lasing and RF illumination of targets by manned simulators is managed and distributed by Common Services and replicated in the local Synthetic Battlespace of each interested participant. Entity expendables (chaff, flares, etc) are modeled as events from their respective simulation to provide the “seed” or “initiating” data required for Ordnance and Signature Services to perform geometry and signature calculations necessary to determine their effectiveness on sensors.

All emissions emanating from manned simulators or SAF entities and individual entity signatures are managed and processed by Emission and Signature Services for commonality and correlation across the Synthetic Battlespace. Applications requiring emission data from remote entities for the purpose of determining detection, electronic surveillance, or jamming must obtain such data from their local common emissions database. A common emissions database is replicated across all nodes to maintain correlation and consistency across the Synthetic Battlespace. Manned and SAF simulators only need to model local avionics sub-systems in this architecture while the Common Battlespace Services provides all necessary emission and signature data to the manned and SAF simulators from the Synthetic Battlespace. When entity behaviors dictate a change to the emission characteristics of sensor and countermeasure systems, the owning application provides the input state command to the Common Synthetic Environment with basic emissions information (frequency, power, PRI or PRI range, mode, sensor position initialization data, azimuth and elevation relative to the sensor’s platform), and a mode index reference to the emission database. The emissions database contains sensor scan pattern and complex or high iteration emissions data that are referenced by local Common Battlespace Services for regeneration and consumption by the local Application. This architecture minimizes the passing of enormous quantities of data caused by the interactions between automated and semi-automated electronic and electronic warfare systems by the use of regeneration of pre-known mode, scan, and emissions data locally. All instances of the Synthetic Battlespace process results locally, using identical algorithms, emission data, signature data, and natural environment data. Signals, jamming, expendable decoys, platform geometry, and natural environment clutter is accounted for in the data which Emission Services provides to the sensor and seeker models in the Component and/or Application layers. The Component and Application layer models are responsible for assessing the effects on sensor and seeker models and providing behavioral responses. The Emission Services provides algorithms to determine the emissions of interest for each Application, and inform the Application of those emissions whose signature meet filtering criteria. Occulting, LOS, and clutter processing of entities within the Synthetic Battlespace occur for emissions or signatures of that particular entity. Terrain calculations between entities (occulting, HAT, surface clutter) and emission path attenuation are performed within the Common Synthetic Environment using a local replication of the gaming area common terrain and atmospheric database. The Common Battlespace Architecture provides for correlated and consistent interactions between entities regardless of origin (manned simulator or SAF).

CONCLUSION

The Common Battlespace Architecture provides many advantages over embedded and independent SAF implementations including:

1) Scalability in a multi-processor/multi-computational system environment.
2) Common synthetic environment (Synthetic Battlespace) allowing all simulations to process the identical data with common algorithms.
3) Applications (manned simulator and SAF) focus on core behaviors and crew interfaces

instead of implementing unique tactical and natural environment simulations.

4) Fidelity of tactical and electronic warfare environments will remain consistent across all participating simulators and supports high fidelity OFP and EID driven systems for data input/output.
5) Consistent correlation between manned and SAF simulators with respect to terrain and atmospheric natural environments.
6) Local regeneration of extended emission data reduces network bandwidth congestion over LAN and WAN topologies.
7) Reduced latency of tactical entity and emission interactions due to locally replicated Synthetic Battlespace in the Common Synthetic Environment that is accessed by manned simulator and SAF simulations without need to query resources over the network.

The Common Battlespace Architecture allows individual manned simulators and SAF applications to concentrate on their specific domain behaviors while gaining a common simulation architecture capable of providing network connectivity, interoperability, and a common Synthetic Battlespace in which to operate. This common architecture will create and manage the environment where all elements within the Synthetic Battlespace play by the same rules, data, algorithms, and processes to provide a fair-fight for all participants, consistent outcomes for given situations, minimize unbelievable simulation events, reduce LAN/WAN bandwidth requirements and minimize LAN/WAN latency through the use of local regeneration of complex, interdependent, and high iteration data.

REFERENCES

AFIWC/SMM (Sept 15, 1999). *Software Design Document for the Distributed Information Warfare Constructive Simulation*.

AFIWC/SMM (Sept 17, 1999). *Software User's Manual for the Distributed Information Warfare Constructive Simulation*.

Amherst Corp, *RF Simulation Systems (CEESIM and RISS)* http://www.amherst.com/products/rf_simulation_systems.html

Effectiveness of Navy Electronic Warfare (ENEWS) http://heron.nrl.navy.mil/contracts/99baa/923.htm

Excalibur Systems Limited. *Virtual Integrated Electronic Warfare Simulator (VIEWS)* http://www.excalibur.com

Falash, M., Holcomb, J., & Pollak, E., "Evolution of a Scaleable API Based Simulation Framework," SESS 1998, New Orleans, La, May 1998.

IEEE Std 1278.1 (1995). *IEEE Standard for Distributed Interactive Simulation – Application Protocols*

JSAF. *Joint Semi-Automated Forces Documentation.* http://www-code44.spawar.navy.mil/advtech/navysf

Moya, P. (1997). *ADA and ModSAF: An overview of US Army Air Defense Capabilities Air Defense Operations and Capabilities* http://www.fas.org/man/dod-101/sys/missile/docs/blueada.html

OneSAF. *One Semi_Automated Forces.* http://www.onesaf.org

Pike, J. (1997). *Improved Many on Many [IMOM]* http://www.fas.org/irp/program/process/imom.htm

Pollak, E., & Falash, M. (2000). *A Common Architecture for Crew Skills & Tactical Training System*. Paper presented at the International Conference, Exhibition and Network For Training, Education, and Simulation (ITEC).