

Data Enrichment and Enhancement in Support of Interoperability Between Combat and Intelligence Simulations

Lou Ford and Roger Smith
Titan Systems Corporation
12424 Research Parkway
Orlando, FL 32826
lford@titan.com
rdsmith@titan.com

ABSTRACT: *When multiple simulations are brought together to jointly model a problem, there is a natural and inescapable gap between the information represented in one model and the information needed by another. Traditional interoperability mechanisms have focused on integrating simulations that possess homogeneous models of the domain. These mechanisms do not provide standardization in bridging the gaps that exist between heterogeneous representations of the domain.*

Within existing DOD simulation systems, one of the most common gaps between heterogeneous models has been between those that view the world as a place for physical combat and those that view the world as a place for information exchange and collection. For at least a decade, simulation projects have wrestled with this issue of interoperability. The general solution has been to create mechanisms ranging from the very simple to the very elaborate for generating missing data based on information that is available from other simulations and sources.

This paper describes both historical and current leading edge techniques for bridging the combat-to-intelligence simulation conceptual modeling gap. It describes the techniques first put in place in the late 1980's to join the TACSIM intelligence simulation to the Corps Battle Simulation and JANUS. It then goes on to describe extensions of this work used for models like FIRESTORM, CSTAR, IEWTPT, and intelligence members of the JSIMS Alliance.

1. Introduction

The Intelligence and Electronic Warfare Tactical Proficiency Trainer (IEWTPT) provides the ability for intelligence system operators in tactical units to train while sitting at the controls of their real-world system. It interfaces the combat environment generated by WARSIM, Combat Training Centers (CTCs), and JCATS with high-fidelity representations of intelligence sensors and processors. A fully integrated IEWTPT exercise includes systems and personnel that are training in constructive, virtual, and live modes. Using this training system, MI soldiers can conduct analysis, write and file reports, change tasking of simulated sensor platforms, and do any of the tasks that could be with the IEW system when receiving real world data.

IEWTPT will provide for individual, crew and collective training for the Intelligence Battlefield Operating System (IBOS). An IEWTPT model that will provide a realistic picture of the battlespace on their IEW system will stimulate soldiers manning the IEW systems. The collection manager will be able to task his various sensors and collection assets in order to meet the commander's objectives for his intelligence assets. The intelligence battlestaff will

conduct realistic intelligence fusion and analysis using reports from the soldier/analyst. Commanders will make decisions based on realistic intelligence products provided by his staff and not intelligence products provided by computer simulations. [1]

The IEWTPT system will complete the interaction between battle command staff and the IBOS. The system will replicate the interdependent processes and stresses encountered in wartime and provide opportunities that force the commander to develop and manage intelligence requirements. IEWTPT will drive intelligence soldiers and systems to answer operational needs and support staff/unit sustainment training.

The IEWTPT consists of four major components:

- Technical Control Cell (TCC),
- Target Signature Arrays (TSAs),
- constructive simulations, and
- training range instrumentation interfaces.

“The TCC controls all communication between the constructive simulation and the IEW processing and collection systems. Control functions of the TCC include: segregating and linking the tactical IEW processing systems to provide individual, collective, and unit level training; collecting training data for the

After Action Review (AAR); and providing to the constructive simulation the status of the tactical IEW system. The TCC will enhance the constructive simulation runtime state variables to provide the data to the TSAs. The constructive simulation is the scenario driver for the IEWTPT training system and provides the runtime state variables and the synthetic environment. The training range instrumentation interface will be used at the NTC, the Joint Readiness Training Center (JRTC), Ft. Polk, LA and the Combat Maneuver Training Center (CMTC), Hohenfels, Germany to provide data to the TCC from live, instrumented units. There is also a requirement to provide an interface to the Homestation Instrumentation System (HITS) when it is fielded.” [1]

In an environment driven by a combat-specific simulation system like WARSIM or JCATS, there are many intelligence and electronic warfare-specific variables that are not provided by the driving simulation. When this occurs it is necessary for the intelligence simulation to add details to each object and interaction to make a determination about its detectability or its electronic impact on the battlefield. This process has come to be referred to as enhancement. In this paper we describe the enhancement process being used to connect IEWTPT to WARSIM, JCATS, and the CTC's. We are also providing background information describing how this process has been handled on previous systems.

2. Background

Networked heterogeneous simulations have practiced enhancement since the beginning of multi-simulation interoperability. We will describe the evolution of this process in systems that are considered to be legacies of the current IEWTPT and WARSIM systems.

2.1 CBS and the TACSIM

The Corps Battle Simulation (CBS) and the Tactical Simulation (TACSIM) exchange data via the Aggregate Level Simulation protocol (ALSP) as members of the Joint Training Confederation (JTC). As an aggregate-level combat simulation, CBS provides information about the state and actions of a unit containing multiple vehicles. This data is largely limited to that necessary to move, detect enemy units, and engage them. No information about the physical appearance of each vehicle is available and this type of information is essential to allow an intelligence simulation to do imagery intelligence (IMINT) and to report on the vehicles that are detected. Neither does

CBS provide any information about RF emissions of radar equipment associated with air defense and other similar units, which makes electronic intelligence (ELINT) impossible. Within the simulation, CBS units do not exchange radio messages, which makes them impervious to communications intelligence (COMINT).

To collect intelligence about CBS units, TACSIM must enhance the state and events created by CBS to add the necessary characteristics. This data is created prior to an exercise and stored in a series of tables for runtime access. Each record is keyed to the values of specific attributes provided by CBS. As those values change during an exercise, different enhancement records are matched with the unit and the intelligence signatures of the unit change as well. The dominant CBS variables that are used as indices into enhancement records are unit type, vehicle type, echelon, activity, and time in activity.

Imagery Intelligence. For imagery intelligence, vehicle type and activity are sufficient for creating a signature of the vehicles. Vehicle type becomes an enumeration similar to those used in DIS PDUs and is mapped to a series of descriptive phrases of the vehicle. Since the TACSIM sensor model differentiate between four different levels of detection, the enhancement data for each vehicle must include descriptions of each vehicle as it would be reported at four different levels of detection. For an armored vehicle like a tank these four levels may be: possible vehicle, tracked vehicle, tank, and T-80 tank. The physical capabilities of the sensor under the given environmental conditions determine the National Imagery Interpretation Rating Scale (NIIRS) and National Radar Interpretation Scale (NRIS) levels that correspond to the four levels of detection.

Signals Intelligence. For signals intelligence all of the CBS variables listed above are used to identify the specific type of radar equipment that would be associated with the aggregate unit and to estimate its operational state. The activity being performed by the unit and the time that it has been in this activity are used to determine whether the radar equipment would be emitting and which mode it would be operating in. Once this is determined, the unit type and echelon are added to determine the operational frequencies associated with the radar equipment. A random number selection is used to select the radar frequency from a trapezoidal distribution covering the operational frequency band. Other RF values are drawn from the prepared tables to allow TACSIM to create Tactical ELINT reports with all of the necessary fields populated.

Communications Intelligence. COMINT is the most difficult for of intelligence to generate realistically using enhancement. This requires all of the steps described in the SIGINT section, plus the selection and creation of internal content. Internal phrases are prepared by hand and include “punch outs” to allow the insertion of dynamic data from the detected unit. These punch outs allow the software to insert data like the name of the unit, its location, activity, echelon, parent unit, level or health, and just about any other variable that can be drawn from the state data of a CBS unit. The library of phrases is indexed by the CBS unit type, echelon, activity, and time in activity. The latter provides unique phrases that will be released a unit remains in the same state for a long period of time.

TACSIM contains the most extensive and well-developed enhancement techniques of any operational intelligence simulation. Many other systems have implemented one or two of the techniques pioneered by TACSIM during the 1990’s.

2.2 FIRESTORM and HRSS

FIRESTORM and the High Resolution Simulation System (HRSS) were originally developed by the simulation center at Fort Huachuca. There focused on providing data that could be presented to the user in a more detailed form. Receiving the aggregate unit information from CBS, these systems prepared it so that the vehicles could be represented in a 3D stealth display representing the cameras aboard a UAV and as moving target indicator data captured by the J-STARS system and its corresponding Ground Station Modules (GSMs).

Both FIRESTORM and HRSS focused on disaggregating aggregate units into individual vehicles and providing basic data about the identity of the vehicle. The UAV camera required mappings to enumerations that could be used to select 3D models of each vehicle for display on the terrain. J-STARS models required the location of each vehicle, its type (wheeled or tracked), orientation, and speed. This data was used to create moving map of dots that revealed the movements of large units across the battlefield.

The software and models developed under FIRESTORM and HRSS have evolved into the Combat STAR program and have been integrated into TACSIM. They no longer exist as independent systems.

2.3 TACREP Generator

The TACREP Generator (TRG) project attempted to provide more realism in the creation of phrases included in COMINT reports. It used some of the same techniques as those described in the TACSIM COMINT section above, and added one new technique. The TRG database of phrases was created such that multiple phrases could be joined together. The intention was not only to report on the activities of a single unit, but to associate those activities with other units that were geographically or organizationally close to the detected unit. Therefore, a report might describe what a number of units were all doing together.

TRG tried to add one more level of complexity and realism to COMINT reports, making them read more like what a human analyst would send to his superiors. TRG has also been incorporated into TACSIM and no longer exists as a separate system.

2.4 JSIMS Intelligence Simulations

Within JSIMS there are a number of intelligence models that will be implementing the next generation of enhancement. The combat models within JSIMS continue to focus on physical state and activities with little information about the intelligence signatures of the vehicles in the aggregate units. Simulation systems like the JSIMS Signals Intelligence Simulation (J-SIGSIM), National Simulation (NATSIM), Joint Tactical Intelligence Model (JTIM), and Deployable Intelligence Simulation for Collaborative Operations (DISCO) must develop enhancement techniques that build upon the JSIMS FOM. Details on the inner workings of those systems are not currently publicly available.

3. Enhancement

One of the major benefits of IEWTPT is that it enhances data received from constructive simulations. This enhancement comes in many forms. It includes turning on or off the lights on vehicles and buildings according to the simulation time. Combat vehicles may also have their blackout lights on, and civilian vehicles use normal headlights. Enhancements at this level are determined by scripted database entries.

Enhancements extend to SIGINT, MTI, and MASINT signatures as well. Interactions such as radio messages will have any missing SIGINT specific attributes added. These can include frequency, modulation, bandwidth, and emitted

power. Entities that are potentially detectable by MTI or MASINT include radial velocity, vehicle type (wheeled or tracked), vehicle composition, mass, velocity, and infrared signatures. The depth and accuracy of the enhancements is only limited by the attributes included in the IEW FOM, and information put into the enhancement database. Therefore, it is possible to expand upon the list to support future connections to new types of intelligence ground stations.

3.1 Objects

In HLA simulations there is a distinction between objects, which have state, and interactions, which do not. Enhancement of objects is considerably cleaner than enhancement for interactions. As information becomes available to populate an object's enhanced attributes, an update to the object is published. Simulation participants that are interested in those attributes receive them as they would other attribute,

and participants that are not interested do not receive them.

For example, when a tank is created its standard attributes are published in the - attributes like position, location, and velocity. A workstation performing enhancements on tanks receives this creation or update of a tank instance and performs a database look-up with the indices corresponding to the criteria for tank enhancement. Indices for determining the state of the "blackout lights" might include the time-of-day and the velocity of the tank. At nighttime a tank with a velocity greater than 2 mph, would cause the enhancer to set "blackout lights" attribute to "on". For exercise participants like intelligence collectors who care about the lights, the status of the blackout lights is received just like any other attribute the tank possesses. Table 3.1 provides some of the combat objects from the JSIMS FOM that are being enhanced in the IEW FOM.

Table 3.1. Enhanced version of JSIMS FOM objects' visual attributes as they appear in the IEW FOM

Objects	
BaseEntity.PhysicalEntity.Platform	BaseEntity.PhysicalEntity
AfterburnerOn	EngineSmokeOn
AntiCollisionLightsOn	FlamesPresent
BlackOutBrakeLightsOn	SmokePlumePresent
BlackOutLightsOn	
BrakeLightsOn	BaseEntity.PhysicalEntity.Lifeform
FormationLightsOn	FlashLightsOn
HatchState	StanceCode
HeadLightsOn	
InteriorLightsOn	BaseEntity.PhysicalEntity.CulturalFeature
LandingLightsOn	ExternalLightsOn
LauncherRaised	InternalHeatSourceOn
NavigationLightsOn	InternalLightsOn
RunningLightsOn	
SpotLightsOn	BaseEntity.PhysicalEntity.Sensor
TailLightsOn	AntennaRaised
	BlackoutLightsOn
	LightsOn
	InteriorLightsOn

3.2 Interactions

Since interactions lack state (i.e. they don't persist in the battlefield the way tanks, trucks, and radar equipment do), their enhancement requires a fundamentally different approach to providing parameters not available when the interaction is originally published. This problem can be divided into two categories. The first approach publishes follow-on or supplementary interactions to add data

attributes data to the original interaction. The second approach intercepts interactions and performs

enhancements/additions before re-publishing them. This requires the use of pairs of interaction classes. The original class is enhanced and transformed into a second class form for republication.

Table 3.2 Sample of enhanced JSIMS FOM interactions as they appear in the IEW FOM

Interactions	
event.message.call_for_fire	event.message.salute_report
event.message.iir_c100	event.message.intcollnom_d170
event.message.iir_ipir_supir_c100	event.message.intrep_c110
event.message.oprep1_f002	event.message.intreq_d101
event.message.oprep3_c487	event.message.intsum_g131
event.message.order_a423	event.message.ri_f014
event.message.reccexrep_c101	event.message.mastr_s304
event.message.tacelint_c121	event.message.rri_f015
event.message.tacrep_c111	event.message.ussid_tacelint
event.message.salute_s303	event.message.ussid_tacrep
RadioSignal.ApplicationSpecificRadioSignal	RadioSignal.EncodedAudioRadioSignal
RadioSignal.DatabaseIndexRadioSignal	RadioSignal.RawBinaryRadioSignal

manage a system for linking the enhancement interactions to the original interactions, and all other

Follow-on interactions

Performing enhancement via supplementary interactions can follow one of two courses. The first is to reproduce the original interaction with the extra parameters from enhancement algorithms. This is the simplest approach and is almost identical to the approach for enhancing objects. The workstation that performs the enhancement simply subscribes to the interactions it will enhance, and upon receipt of those interactions works through its algorithm and publishes the interaction again, this time with the enhanced attributes populated. Participants interested in the enhanced interactions have a slightly more complicated task, since they must now expend some effort to distinguish between enhanced and non-enhanced interactions. While this is the simplest approach algorithmically it consumes more network bandwidth, since all the non-enhanced parameters of the original interaction are pushed across the network a second time in the enhanced interaction.

workstations interested in the enhanced data must be aware of the linkage and the prescribed method for exploiting it. This linkage must be built into the simulation data itself (that is, it must be expressed in the FOM). Additionally, workstations interested in the enhanced data must subscribe to both interactions (the original and the smaller supplementary interaction) in order to have a complete picture of the situation.

A second course is to produce a second interaction, linked to the original, with only the additional parameters in it. This approach improves upon the above problem of network bandwidth consumption, at the expense of a slightly more complex algorithm. The workstation performing the enhancements must

Intercepted interactions

A distinct approach to the problem of enhancing interactions is to intercept them and republication the interactions in a separate federation. This requires the use of a dedicated federate between the federate originating the data and the federate consuming the data. This is the approach being used on IEWTPT. In fact, IEWTPT uses a gateway federate to bridge two different federation with two different, but related, FOMs. Interactions subject to enhancement are created on a separate federation from the main federation for the simulation. The workstation performing the enhancements subscribes to the interactions on this second ancillary federation and publishes them with the enhanced parameters on the

main federation. Thus, there are no additional interactions published on the main federation. If both federations are maintained on the same network then this approach suffers from the same overhead issues as the first “republish” approach described above. However, since it involves an entirely separate federation, a separate network can be established all the way down to the physical/hardware level if necessary to preserve bandwidth for each federation. The main drawback to this approach is that both the creator of the interactions and the enhancer of those interactions need to be aware of the additional federation and what is published and subscribed on it. A less significant disadvantage to this approach is that it is not easily inserted or removed from the simulation due to that linkage between the creator and the enhancer roles.

In addition to enhancing interactions that have already been published in the simulation there is the question of inserting, or injecting, interactions into the simulation. This type of enhancement could be considered “inferred” interactions. Triggers for this type of enhancement are difficult to establish. For example, objects like tanks and trucks can be moving around the battlefield for a variety of reasons. To infer from their movement the particulars of the order sent from the commanding unit is risky in terms of maintaining correspondence between what is being represented in the created interactions and what is actually happening in the simulation.

Combinations

Some interactions require reference to object state data. For example, if a radio transmission contains the frequency of the emission then the transmitting radio object should be referenced to determine the correct frequency. This runs the risk of “stacking” enhancements (i.e. basing one enhancement upon another). Such stacking does not necessarily constitute an error, but it does increase the risk of non-correspondence between the enhanced view of the simulation and the original ground truth data.

4. Voice Generation

One of the largest gaps in training for the intelligence system operator is tactical training in the target language. There is no easy way to generate high quality text-to-voice translation in target languages. Even in English, the quality of the audio produced by text-to-voice programs is noticeably “mechanical” and often not suitable for use in intelligence training. One possible solution is to provide database indexes to identify message types to TSAs and allow the TSA

developer to determine how to present the data. This is provided in addition to the standard ASCII text string representation of pre-scripted sentences. The TSA could use the index to reference sound files and playback the sound file corresponding to the index. Those sound files can be recorded in the target language of the system operator, providing a more realistic training environment. This solution also allows the TSAs and even the local units to improve upon the database of messages and to tailor it to their specific needs and areas of interest.

5. Traffic Generation

One of the future additions to the IEWTPT system will be automated Traffic Generation. This will add simulated signals emission to the environment, creating many of the signals that would normally be encountered by an operator in the field. This background cultural noise simply does not exist in the constructive simulation. Examples of cultural noise include radio stations, channel markers, Teletype and fax transmission, cell phone traffic, and airport control towers.

Additionally, most constructive simulations omit tactical radio traffic that exists between thousands of individual vehicles on the battlefield. Currently if the battalion commander orders a company of tanks to move to a new location, no simulated radio message is sent. Thus, there is no communication for the intelligence operator to intercept. Instead, a simulated order is passed through the constructive simulation in place of a simulated radio message. The Traffic Generator will create radio transmissions to match the simulation-generated orders that are sent from role player workstations to simulated units. This will provide potential collection targets for intelligence systems.

5.1 How it Works

The Traffic Generator first looks at the filters being used in the system and determines the frequency ranges are being monitored. From these ranges it determines the level of traffic should be generated. If the only frequencies being monitored are in the VHF range, there is no need to generate traffic in the UHF or HF ranges.

The number and types of messages that can appear in an exercise or during real world collection is extensive to say the least. Keeping a list of all possible messages for all situations is impractical. So the Traffic Generator mitigates this problem by using a messaging hash table along with a message

generation database to determine the specifics of the message.

5.2 The Messaging Hash Table and the Message Generation Database

The messaging hash table is used to narrow the message search by looking at the known message components. It differentiates messages by types (e.g. move, attack, defend, road march), sending unit echelon, and receiving unit echelon. A movement message sent from a Battalion to a Company would be different from a movement message from a Division to a Brigade, and the differences would be even more extreme for different message types. By following these levels down through the hash table, it eventually leads to the message generation database, which would contain the appropriate message and the index into the database.

5.3 Tactical Message Traffic

Tactical message traffic primarily works by examining the interactions passing between entities and units, and generates a database index to reference the appropriate radio message. For example, when a JSIMS role player issues an order to Computer Generated Forces (CGF) that they command, an *event.sim_order* interaction is sent out to all units and equipment groups involved. This command is not carried in the form of a radio message and does not involve any message-like text. The Traffic Generator will monitor these interactions and use them as triggers to generate radio message indexes. By examining the order of battle information, the command structures between objects, and the orders sent in the *sim_order*, the Traffic Generator will determine the appropriate message to be sent.

5.4 Cultural Message Traffic

Cultural Message Traffic is created differently from the Tactical Traffic. Cultural Traffic does not rely on interactions or entity updates as triggers for performing enhancement. To create Cultural Traffic the Traffic Generator first looks at the filters describing frequencies being monitored [4]. It then determines the type of cultural noise would fall within those frequency ranges. If a TSA is collecting radio traffic, and the frequency range covers the AM or FM radio bands, the Generator will create radio station traffic such as music, sports, and news reports. If a TSA is focused on a range of frequencies that includes aircraft control towers, then the appropriate aircraft control messages will be generated. The prime purpose for this type of traffic generation is to

add to RF clutter that a SIGINT operator typically finds in a real world environment.

6. Challenges

When two simulations that have different levels of resolution are joined together, the addition of information as described in this paper can create an issue involving correlation with the original simulation. Data that is added by IEWTPT must correspond to related events and states that will be generated by WARSIM, WIM, JCATS, and/or JANUS. Those simulations do not track information about the RF emission of a radar system. Therefore, the RF signature that is generated must be consistent with the operational state of the physical object. The identity, activity, attrition levels, and rule of engagement of the unit owning the radar may determine these emissions. In some cases, the appropriate RF signature is not directly derivable from data about the unit. The type of terrain being traversed, status of parent units, recent orders received, and damage levels may call for emission status in the real-world that is difficult or impossible for a simulation system to predict. When this occurs it is possible for the simulation to generate emissions when the unit should remain silent.

Another problem arises when multiple simulations in a federation are enhancing data from a single object. Since each simulation is an independent system that is operating from their own rules and databases, it is possible for each to choose a different value for the same unit. For example, an airborne ELINT system may add a bandwidth of 10 MHz to a unit while at the same unit receives a bandwidth of 20 MHz from a simulation controlling ground based detector. Reports on this unit from the two systems will be inconsistent even though they are referring to the same target unit and emissions at the same time. Coordinating the set of rules and database values that are used by the simulations doing the enhancing can eliminate some of these inconsistencies. However, we have learned through experience that the rules and data can be implemented in different forms that makes it impossible to correlate them. As an example, a series of specific operating frequencies may be selected for radar using a random number generator. It is usually impossible for two independent simulation systems to generate the same random number to make this decision.

7. Conclusion

Enhancement is neither easy nor cheap. It requires significant data repositories, both in knowledge

databases providing rules for enhancements, and in content databases providing data for enhancements. Significant processing can also be an issue, since the criteria for providing enhancements (particularly in creating interactions not provided from the constructive simulation) can be complex. Accuracy and Consistency are always issues with other simulations, since no two simulations view the real world from exactly the same perspective. Despite the difficulties, enhancement warrants the necessary investments since it enriches the virtual environment, improves the quality of training delivered, and expands the training audience, allowing tactical equipment operators to participate in large-scale constructive and live simulations where they would otherwise be excluded.

8. References

- [1] May, Phillip. (March, 2002). "Management Overview of the Intelligence and Electronic Warfare Tactical Proficiency Trainer". *2002 Spring Simulation Interoperability Workshop*. Orlando, Florida.
- [2] Williams, Marcia. (March, 2002). "The Development of an Intelligence and Electronic Warfare FOM to Bridge Constructive, Virtual, and Live Simulations". *2002 Spring Simulation Interoperability Workshop*. Orlando, Florida.
- [3] JSIMS. "The Joint Simulation System Federation Object Model". Version 7.1. November 2001. <http://www.jsims.mil/>
- [4] Stoner, Don, et al. (September, 2002). "Virtual Sensors in a Constructive and Live World". *2002 Fall Simulation Interoperability Workshop*. Orlando, Florida.

Author Biographies

LOU FORD is a Software Engineer for Titan Systems Corporation working on modeling and simulation technologies that bridge the virtual, constructive, and live domains. He has worked on numerous simulation, space, and missile programs and is currently leading a team of software developers on the Intelligence and Electronic Warfare Tactical Proficiency Trainer (IEWTPT). Lou serves on SISO's IO-ISR Committee.

ROGER D. SMITH is a Vice President of Technology for Titan Systems Corporation working on next-generation simulation applications and distributed computing technologies. His most current work has been studying information operations and counter-terrorism as well as the development of new intelligence simulations. He is the creator and instructor for a series of military simulation courses that have educated hundreds of simulation professionals. He is also the Area Editor for Distributed Simulation for *ACM Transactions on Modeling and Computer Simulation* and is actively involved in promoting the expansion of the simulation profession.