A CASE STUDY ON MODEL INTEGRATION, USING SUPPRESSOR

Gregory L. Douglas
L-3 Communications Corporation,
Link Simulation and Training Division
PO Box 1237
Binghamtom, NY 13902
607-721-5879
gldouglas@link.com

Keywords:
CGF, DIS, HLA, SBA, SDG, Suppressor

Abstract

In an effort to create a reusable Computer Generated Forces (CGF) model that would be useful in supporting Simulated Based Acquisition (SBA) environments, an opportunity was presented to modify interfaces to Suppressor in order for it to operate in such an environment. Using real-time modifications to Suppressor as a baseline, it was desired to further create a CGF that would support the integration of multiple models and simulations. The desired outcome was to develop a model that would allow a combination of other models and simulations to play together, sharing data and commands, to represent one entity in Suppressor (i.e., an aggregate of the parts simulated in various simulations and Command, Control, Communication, Computers, and Intelligence (C^4I) systems). At the same time, the infrastructure of this system had to be flexible to the point that no specific external model and no specific number of external models had to be present in the exercise in order for the entire entity to exist. It was determined that a flexible system such as this would be beneficial to those pursuing SBA activities because it would provide a means of piecing together a variety of systems until the user came up with a workable solution that was capable of meeting all of their design goals. This paper will give a brief overview of Suppressor and the underlying real-time infrastructure. It will explain the different variations of subsystems that can now be used to create a "configurable" entity within Suppressor, describe how this type of approach could be beneficial in a dynamically changing SBA environment, and present major lessons learned.

Author Biography

Gregory Douglas is a software engineer at L-3 Communications Corporation, Link Simulation and Training Division. He has a Bachelor of Science Degree in Mathematics and a Minor in Computer Information Systems from Keuka College, Keuka NY. He has two and a half years experience in the simulation industry and has been working with Suppressor for two years. Greg has valuable experience in both incoming and outgoing interfaces to Suppressor, as well as experience making desired local enhancements to the Suppressor model. He is also very familiar with Distributed Interactive Simulation (DIS) protocols through the use of the Suppressor DIS version 2.04 interface.

A CASE STUDY ON MODEL INTEGRATION, USING SUPPRESSOR

Gregory L. Douglas
L-3 Communications Corporation,
Link Simulation and Training Division
PO Box 1237
Binghamtom, NY 13902
607-721-5879
gldouglas@link.com

Keywords:
CGF, DIS, HLA, SBA, SDG, Suppressor

INTRODUCTION

The Department of Defense (DoD) is committed to providing faster required war-fighting systems to the Military Services at a lower cost. To accomplish this, a key area under acquisition reform is the increased use of Modeling and Simulation (M&S) technologies across all phases of defense war-fighting systems. A new approach to acquiring these systems is termed Simulation Based Acquisition, or SBA.

What is SBA? To work toward a common understanding, the DoD published a 1998 report by the Defense Systems Management College to provide definitions relative to SBA. According to the report, SBA is "... an iterative, integrated product and process approach to acquisition, using modeling and simulation, that enables the warfighting, resource allocation, and acquisition communities to fulfill the warfighters material needs, while maintaining Cost as an Independent Variable over the system entire lifecycle and within the DoD system of systems" (Johnson et. al (1997)).

The main feature that would be beneficial to the SBA capability described above is the ability to provide one simulation that would offer an environment that supports the integration of any combination of external models and simulations. A means of analyzing the effectiveness of each environment configuration would also be a desirable feature. Since Suppressor already provides this analysis capability, the approach would be to modify Suppressor to support model integration by allowing any piece of data for a player to originate from any external simulation. At present, no one CGF provides all of the needed capability to support the range of these models and simulations across the wide fidelity levels required for full SBA. The Suppressor concept allows external simulations to have "plug and play' capability to support full SBA requirements. Of course, this would have to be done without compromising any of the functional or analytical capabilities of Suppressor.

The method used to meet the above goals was to modify the interface architecture to Suppressor and, thereby, provide "configurable" entities (i.e., entities whose different "pieces" could come from either Suppressor or an unnamed external model or simulation). Through real-time interfaces, Suppressor subsystems could be replaced with subsystem models of a higher fidelity. This replacement was to take place not by putting the new model into Suppressor, but interfacing the new model into Suppressor over a Distributed Interactive Simulation (DIS) or High Level Architecture (HLA) distributed network connection. This configurable approach would then allow the user to easily choose how entities are configured (i.e., what model(s) or simulation(s) will control the entity's movement, Command and Control system, sensors, and weapons), from one run to another, without sacrificing any of the capabilities of Suppressor or changing the Suppressor model.

General Overview of Government Furnished Suppressor

Suppressor is a DoD, analytical, event-stepped, many-versus-many threat model. Its traditional use has been in the analytical community to do Measure of Effectiveness (MOE) and Cost and Operational Effectiveness Analysis (COEA). It simulates human behavior, sensors (infrared, electro-optical, radar, and radar warning receivers), radios, jammers, weapon systems, and movement systems. The human behavior algorithms are separate from the physical systems. They execute database-defined tactics for resource allocation and movement evaluation. Resource allocation entails picking a target and associating it with specific sensor and weapon systems to form an engagement. Movement evaluation involves picking the appropriate movement plan as well as determining whether or not to perform terrain maneuvering and/or threat avoidance. After the desired movement is determined, a separate set of route planning algorithms, which take into account physical limits such as minimum and maximum speed, minimum and maximum altitude, and minimum turn radius, combine the desired avoidance technique to determine the actual path that will be followed.

The human behavior algorithms do not make decisions based on ground truth, but on perceived truth. A player's perceived truth is obtained by noticing and mentally digesting what is detected by the player's sensors or what has been communicated to that player from another. Command and control structures, as well as a player's tactics, help define what will be communicated and to whom. Subordinates can communicate, to their commander(s), targets or players they detect. Subordinates may also have, in their database, logic to act autonomously if communication to their commander continues to fail after a certain number of tries. Commanders can assign a target to a subordinate, control whether a subordinate is allowed to fire a weapon or not, and launch a subordinate (tell it to start movement).

One of Suppressor's most important features is its data capture capability. Via a model input file, the user can choose which types of situations are captured during the run. The captured data can be stored as an American Standard Code for Information Interchange (ASCII) text file and/or as a Suppressor binary file. An analysis process can be run, using the captured data as input, to further filter the output for specific situations, specific players, or a combination of the two. An analyst can view this data to gain a better understanding of the scenario. Although it is not included in the government furnished Suppressor, there are graphical utilities that have been developed to read the captured data back into Suppressor for a replay or "playback" option. Both of these unique capabilities provide the user with a flexible and analytical option to compliment the threat model.

Changes to Suppressor to Support Real-Time Simulations

As provided by the government, Suppressor runs in non real-time and does not interact with virtual devices. As a result, a real-time interactive shell was added around Suppressor (George et. al (1996), Hanford et. Al, Pope (1999), Pope et. al (1995)). DIS network software was then added to allow Suppressor to communicate with virtual devices. Finally, a Supplemental Data Generator (SDG) was created as an off-line tool to prompt the user for information that is not currently available in the Suppressor model. This user-supplied data could then be read into the real-time simulation during an initialization step.

Supplemental Data Generator (SDG)

The SDG is an Oracle based tool that prompts the user to supply the simulation with data that is not currently available to Suppressor. The concept of the tool is that data for a specific Suppressor player name and number combination is stored in library files. The library files are searched each time the tool is run. If any information is found to be missing for that particular player, the user is lead through different Oracle screens that prompt for the necessary data to be input.

One example of SDG originated data is the DIS networking information for each player in the scenario. The user is prompted for the necessary DIS entity information (kind, domain, category, etc, etc….). This data is then stored into the necessary data structures during real-time and is referenced when preparing to publish any entity information onto the DIS network. Likewise, DIS information for sensors, communication transmitters, and weapons in the scenario is originated within SDG.

Another use of SDG prompts the user for the player types in the scenario that are to be externally controlled. This user-entered information is also stored in a data file that is read during real-time initialization. All external players are "ghosted" in Suppressor and rely on the network data for that player.

Suppressor Real-Time Shell

The Suppressor real-time shell constrains Suppressor to maintain time based on a real-time clock instead of allowing it to run as fast as a computer will allow. The initialization task is also contained in the real-time shell. This initialization task is run prior to Suppressor's initialization and is used to read in data supplied by the SDG. Another aspect of the real-time shell is the export data handler. The export data handler exports data out of Suppressor and stores it in a "generalized interface." This interface has been generalized to support the ease of integration with virtual devices. It contains positional, active sensor, passive sensor, communication, jammer, and weapon data. It is important to move all of this data to the generalized interface since it provides a means of quickly accessing Suppressor data without software having to search through Suppressor's data structures for entity information. The external data initialization package provides routines to create an external player, within Suppressor, as well as set up all internal arrays needed to match up the data in the real-time shell's import area (populated by data pulled off the DIS network) with specific data blocks inside Suppressor. The external data processor takes data out of Suppressor's real-time shell import area and puts it into the government's Suppressor data structures. This step is necessary so that the Suppressor model will be able to make decisions and calculate reactions based on externally fed data. The positional data is used to create movement paths within Suppressor. The external communication-system data and radar data are put into the appropriate corresponding Suppressor data blocks for that particular system. Finally, external weapon fires are also pulled into Suppressor, and, if within range of a Suppressor controlled target, Suppressor algorithms are processed to determine the damage done to the target.

Under this design, all aspects of the real-time shell run on one process and the DIS network software runs on a separate process. However, both executables share much of the same data. This is done via an inter-process, shared-memory, implementation that references a generalized interface to gather data. During each cycle of the software, all new external data is processed within Suppressor and, any information, changed by Suppressor that needs to be made public, is put on the DIS network.

What Was the Desired Outcome?

With the previously discussed software design, it was then desirable to make the necessary modifications to support a predefined SBA environment. As required for a specific SBA experiment, Suppressor would become the "glue" for the SBA environment by having it work in cooperation with other subsystems. It was important to be able to take advantage of all of Suppressor's capabilities, while supporting the integration of a variety of external subsystems.

Design Goals

One of the design goals was to support the integration of multiple models and simulations, as well as provide the capability to support the configuration of any combination of these models. Analysis of given simulations showed that, although there are simulations out there that will simulate every aspect of a war-gaming environment, there is no available model that can simulate all of these aspects to the required degree of fidelity. Therefore, it was highly conceivable that any SBA environment would consist of a hybrid of existing simulations in order to reach the desired outcome (Ewen et. al (2000)).

This first design goal naturally lends itself to the second – the need for overall system analysis of effectiveness. With the use of multiple models and simulations, combined with the dynamically changing configuration of these models (from one run to another), a way of analyzing each configuration to determine the effectiveness of the overall simulation would prove useful. It is conceivable that different criteria could determine the success of a run. Therefore, a proven method of analysis had to be available.

Ground Rules

With these goals in mind, it was decided that Suppressor's real-time interfaces should be expanded to meet the needs of such an environment. Suppressor meets the requirement of the analytical capability. However, architecture changes would be required to support the integration of multiple models and simulations. To do so effectively and successfully, a number of ground rules would have to be followed.

Just as it was important for a variety of models and simulations to participate in the overall simulation, it was equally important that no hard requirements, other than standardized interfaces such DIS or HLA, be placed on any of these "external" systems. This included the flexibility of user defined DIS Protocol Data Units (PDUs).

As would be expected, another ground rule would be that the data, from any and all of these models, would be able to be acted upon by Suppressor just as if it were the originator of the data. To do so, the externally fed data would need to be manipulated by Suppressor's interfaces so that it could be placed into Suppressor's data structures.

One of the most important ground rules that needed to be adhered to was the maintenance of all of Suppressor's current functionality and analytical capability. For example, Suppressor's data capturing ability of scenario events (an important tool for SBA) was a big reason why it was chosen as the "glue" of the simulation. Indeed, collection of external events soon became equally desirable to simply collecting Suppressor events. In turn, interfaces would have to be developed to appropriately inform Suppressor of external activities so its sensors, communication systems, etc., would have the appropriate ability to react and maneuver just as they would if Suppressor had been the originator of the data.

The final ground rule was that the developed solution would have to plan for the future in that the design could not depend on any particular model or number of models. It is not only feasible but very likely that additional features (therefore, additional models) will be necessary in the future. Any modifications to existing code or any re-design to accommodate this would have a vast impact on the overall project. Therefore, the addition of any new models or simulations would have to be as close to a "plug and play" installation as possible. Depending on what was needed, additional coding requirements would be limited to the addition of and integration of additional user-defined PDU structures.

The Solution

Prior to this SBA development, Suppressor's real-time existence allowed an entity to be totally controlled by Suppressor or totally controlled by an external source. The next conceptual step would be to allow pieces of the overall system to be externally controlled. Suppressor would not only have to react to these external "pieces" but control the rest of the entity. Additionally, the determination of which "pieces" would be externally controlled would have to be user controlled, easily modified, and not require software recompilation between runs when external pieces changed.

Configurable Entities

The solution that was decided upon was a concept, within real-time Suppressor, called "configurable entities." Configurable means that the entity would be "configured" by a variety of different sources. These sources could consist of Suppressor and/or any of the external simulations. The best scenario is that Suppressor would control all aspects of the entity with the exception of those systems that are required, or desired, to be of a higher fidelity than what Suppressor can provide. The higher fidelity system would then replace Suppressor's representation of that system. Keeping Suppressor's analytical capability in mind, it would be very easy to determine if this "higher fidelity" system truly produced better results than Suppressor's system. This could be beneficial when deciding what external systems to use. The results from the use of one external system could be compared with the results of a run with another external system. An analyst could then use this data to determine which system produced the best results.

Ease of Setup

Depending on what systems are available and what test or scenario needs to be analyzed, different configurations of the available models are implemented between one execution and another. For example, during one execution testing the results of an external sonar system may be desired. During the next execution, testing the results of an external weapon system may be desired. Somehow, these configurations need to be determined by the user. The solution was a simple executable that could be run at any time prior to scenario execution, that prompts the user for which entities are to be "configurable," and which external systems would make up the configuration. This executable would be generic in that it would not ask for specific models to control the external system. Rather, it would accept a simple yes or no as to whether or not the system is externally controlled. An example of this would be, "which weapon systems are to be controlled externally?" The user would be allowed to make his/her selection(s) from a list of weapon systems for the entity. Suppressor would then look for information regarding this weapon system, to come externally during the simulation instead of handling the decisions for that system itself. Suppressor would not differentiate between multiple external weapon systems. The task would be left up to the user to determine which external weapon system(s) would be allowed to participate in the scenario.

Examples of the Solution

As this "configurable" capability is considered, it is important to keep in mind that any combination of configurations is possible. As a system is designated external, Suppressor will be forbidden to handle the workings of that system and will rely on network data for that system's information. Each possible external system is discussed separately below, but any combination of these systems can be implemented.

External Movement

The first, and perhaps the most complex external system within Suppressor, is the movement of the entity (see figure 1). One of the biggest deficiencies of Suppressor is that it does not represent an entity's movement in the manner of six degrees of freedom. Therefore, the integration of a higher fidelity movement system would probably be desired in many SBA applications. This subsystem interface was developed, to a higher degree than most, in Suppressor. In its simplest form, if the movement of an entity has been designated external, Suppressor will retrieve entity PDUs from the DIS network for this entity from whatever movement system is controlling it. Suppressor will use this data to replace its own movement actions for the entity. As a default, if the movement has been designated external, and no *EntityState* PDUs appear on the network for the entity, Suppressor will take over the movement system for the entity. An example of this setup would be to have another CGF such as Joint Semi-Automated Forces (JointSAF) control the movement of the entity.

An even more complex movement interface that takes advantage of Suppressor's route planning and threat avoidance algorithms was also developed. In order to take advantage of these capabilities, the external movement model would have to possess an interface similar to the one used by the six degree of freedom movement model normally integrated with. This interface would allow the external movement system to operate in manual or C^4I (Suppressor controlled) mode. The C^4I mode is equivalent to an "auto pilot" type model. In C^4I mode, the external system starts the entity at Suppressor's current position, regardless of when the external system is started. To do this, an *ActionRequest* PDU is sent from Suppressor. This serves as an initialization message for the movement model. Suppressor awaits an *ActionResponse* PDU, continually sending out updated *ActionRequest* PDUs until one is received. At the same time Suppressor sends waypoint data out onto the network. This waypoint data contains the latitude, longitude, and Suppressor's suggested arrival time at each of the next two future path points. The reasoning behind this is to give the movement model the ability to take advantage of Suppressor's threat and terrain avoidance algorithms. The movement model takes this input, applies the suggested path points to its own movement algorithms, and attempts to arrive at the suggested points at the suggested time.

In manual mode, Suppressor does not change its behavior. The same requests are sent onto the network. The difference is, the movement model does not respond. It does not start at the current Suppressor position and it does not attempt to follow the suggested navigation route. In either case (C^4I or Manual), Suppressor retrieves current entity data from the network and, if no entity data is on the network, Suppressor defaults to controlling the movement of the entity. This implementation allows the movement model to switch between manual and C^4I mode, at any time during the scenario, without interrupting the entity's movement simulation. In fact, Suppressor does not care about or know in which mode the six Degrees of Freedom (DOF) model is operating.

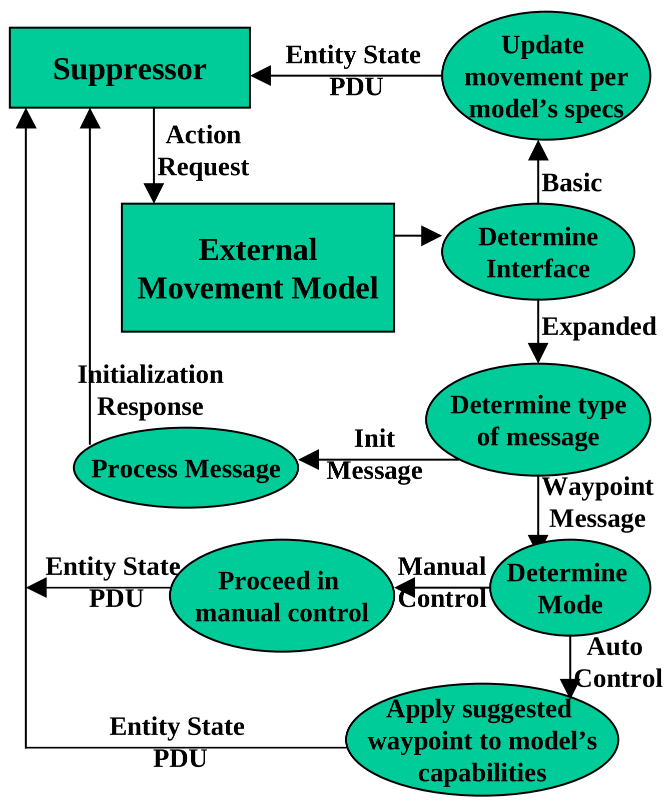

Figure 1. External Movement Model

External C^2 System

The implementation of the external C^2 system is very simple. In normal Suppressor situations, the thinking system digests all relevant data and eventually makes decisions on whether or not to fire and, if a fire event is required, calls the necessary routines to accomplish the task. If the user has chosen the C^2 system to be external, Suppressor's thinking system is forbidden to make the decision to fire. Instead, the decision comes over the network from an external system. This decision is tightly coupled with the status of the appropriate weapon system, but in any case, the necessary course of action is taken to force a fire event to happen; Suppressor may do it itself or it may rely upon an external weapon system to simulate the weapon fire (see figure 2).

External Weapon System

During the pre-scenario executable, the user is presented with all weapon systems found for the entity and is prompted to select which systems are to be externally controlled. As a command to fire is encountered, whether from within Suppressor or from an external source, Suppressor checks its representation of that particular weapon and determines if it is allowed to act upon the command. If the weapon system is designated external, the command to fire is simply ignored to allow the external weapon simulation to handle the command.

Conversely, if the fire command is for an internally represented weapon system, further work is required. Within Suppressor, a weapon fire cannot occur unless the entity perceives the target. Therefore, if the fire command specifies a target, a check is done to see if Suppressor's entity currently perceives the target. If it does, a weapon fire is scheduled to occur within Suppressor. If it does not, normal Suppressor methods of perceiving a target are bypassed, and through the real-time interfaces Suppressor is given a perception of the target. This is subsequently followed by the scheduling of a weapon fire event. If the fire command does not specify a target but rather a specific latitude, longitude, and elevation, a distance check is done to find the closest Suppressor target to the given position. If the closest target is within a predefined 3-D distance, a weapon fire event against the chosen target is scheduled to occur. If not, no further action is performed. Figure 2 illustrates the process for an external C^2 system and an external weapon system.

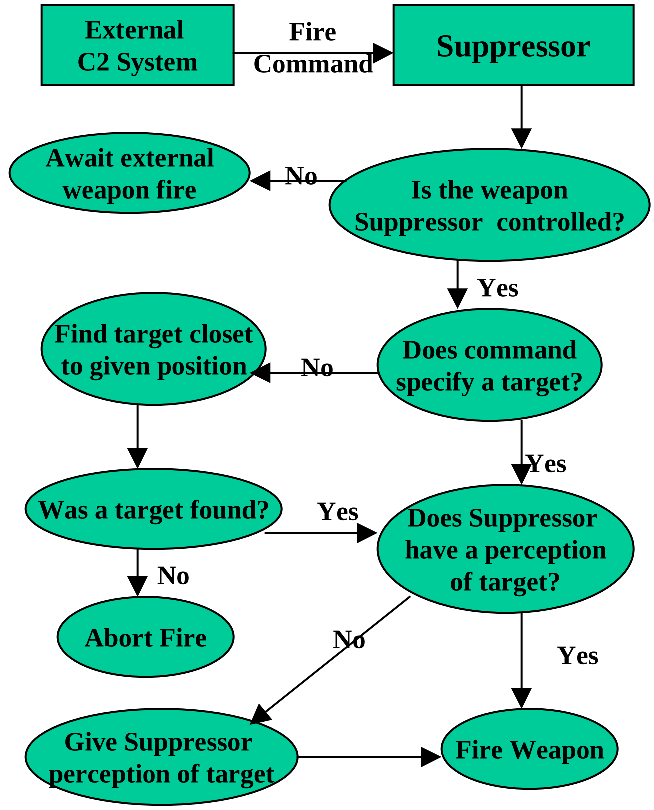

Figure 2. External C^2 System and External Weapon System.

External Sensor Systems

Due to requirements within Suppressor, the notation "external sensors" is a bit of a misnomer. In its truest sense, if a user chooses a specific sensor system to be external, Suppressor will be forced to not acquire any sensing events for the given sensor system. Without sensing events, perception events do not happen. Instead, Suppressor relies upon the external system to communicate sensing events through the use of user defined information in a *Signal* PDU. Since Suppressor does everything based upon perceived truth instead of ground truth, when this *Signal* PDU is retrieved off the DIS network the process of giving Suppressor a detection of the "target" is bypassed.

Instead, through the real-time interfaces, Suppressor is given a perception of the "target." This perceived data is then used within Suppressor during its normal processing. An example is the use of perceived data for threat avoidance and/or movement evaluation. Just as new and updated external sensing events are communicated to Suppressor when sensing updates are no longer found on the network, Suppressor's normal thinking delays are taken into account. When the appropriate time delay has passed, Suppressor will drop the externally initiated perception. For an overview of the entire configurable capability, see figure 3.

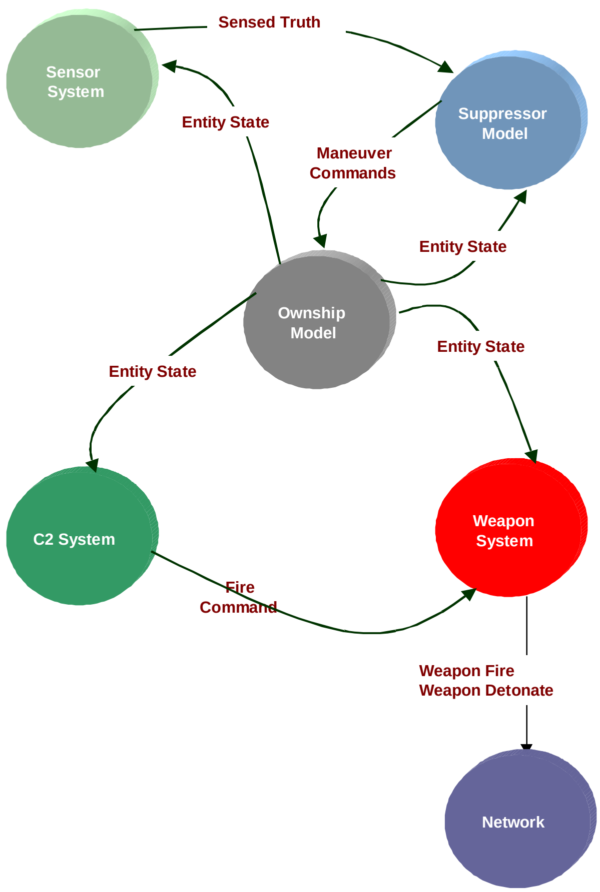

Figure 3. *Overview of the configurable capability.*

Benefits of a Configurable Approach

The use of a configurable approach provides flexibility to the overall simulation. With an attempt to use a single simulation, the SBA investigator is limited by the fidelity to which the chosen simulation operates. In most cases, especially in CGFs, a few subsystems (movement, sensing, etc.) are handled extremely well, while other subsystems (physical models) are not handled as well as possible, handled poorly, or in may cases, not handled at all. With a configurable approach, any subsystem can be upgraded to a higher fidelity as required or desired simply by plugging in an external system. Which external system is to be used is totally arbitrary and left up to the user.

As previously discussed, the number of simulations to be integrated per scenario run is determined by the user. If the results of a particular run are not satisfactory, models and simulations can be added or subtracted as deemed necessary.

Due to the use of Suppressor, the "configured simulation" is even more beneficial. As designed, the SBA investigator is able to take advantage of Suppressor's data capture ability. This, of course, directly impacts the ability to analyze the effectiveness of each configuration. The interfaces were developed in such a way that any external event that happened to or by a "configured" entity is captured in Suppressor's analysis file. This proves to be very beneficial in that there exists a means of comparing the effectiveness of multiple external models that essentially do the same thing. The scenario can simply be run once with each model. An analyst could then view the data captured from each run to do the appropriate analysis.

The use of Suppressor as the "glue" for the SBA system offers an additional benefit. Since Suppressor really models every aspect of an entity to some degree, albeit to a lesser fidelity than desired in many cases, Suppressor can be used as a backup to any and all system models. This is beneficial in two ways. First, debugging and the measurement of effectiveness of an external model are made easier. By allowing Suppressor to simulate all but one system of the entity, there is less confusion on an analyst's side in determining what is actually occurring. For example, since there will be less PDU traffic on the network, a true test of the external model with no other external factors contributing to or disrupting the test is ensured. Secondly, the fact that Suppressor can adequately be used as a backup to many of the desired systems should provide some level of comfort. Using Suppressor to replace subsystems allows the simulation to continue, even if some of the desired subsystems are unable to participate at any stage of the game. The only thing that would be lost would be the level of fidelity, and in some cases, the glitz of the graphics and sound that these external models provided.

Lessons Learned

The fact that the configuration of a scenario was made so easy is desirable for providing a quick and flexible method to modify the configuration of the simulation. However, it is equally dangerous in that it is up to the user to ensure that the models the SBA investigator defined for Suppressor are in the simulation and are actually participating in the scenario. For example, if the user specified that an external sensor model is to be used, Suppressor relies on this to be the case. If the user forgot to start that external sensor model, no detections (or perceptions) are given to Suppressor. Of course, this causes unexpected simulation results. So, the moral is, coordination has to be a top priority when setting up the scenario.

Testing the software became a big factor. Without having all of the possible external subsystems available for testing, it was necessary to create emulators to simulate the data that is normally retrieved from an external system. Although this worked fine for testing incoming interfaces, there was really no way to test the outgoing interfaces from Suppressor until total integration of the product, in the envisioned environment, was achieved. Consequently, additional integration work had to be done to finalize the interfaces.

The fact that every newly designed system has its own unique set of requirements played a major role. Since every simulation system was virtually unique it was never really possible to fully achieve true "plug and play." There always seemed to exist an unforeseen need for specific data that was not thought of before; however, the use of user defined PDUs made it possible to limit the amount of rework. While this approach may have prevented systems from accomplishing everyone's needs, it did provide an amicable compromise between a robust design and schedule impacts.

Throughout the life cycle of simulation, much work has been done to standardize interfaces between entities (the DIS Protocol). However, more work is needed to standardize interfaces within an entity. Successful component based simulation will require this. Our approach was to develop such interfaces ourselves, but the ability to plug and play with different simulations is severely limited by this approach.

Summary

Suppressor, and its involvement in a "configurable simulation," has proven to work very effectively. By inserting high fidelity subsystems as desired, an SBA investigator has neared an "off the shelf" simulation capacity in that the user can pick what models will configure an entity for a specific SBA task. The limitation to this, of course, is the fact that DIS or HLA interfaces have to be available for each model in use; this may require some up-front work on the part of the model being used. However, a system such as this, with Suppressor as the central hub of a simulation, immensely increases flexibility. As long as the "glue" system simulates all subsystems to some degree, it can be used as a backup to any of the desired external models. In addition, as is the case with Suppressor, a way to capture the events that occur within the simulation proved to be valuable when determining the level of effectiveness toward the final goal of each configuration.

In looking ahead, further work has been identified and further work needs to be accomplished before a more "configurable simulation" can exist. The missing links are the ability to incorporate external communication models and external jamming models. So far, the concept has proven successful.

Acronyms

-A-

ASCII	American Standard Code for Information Interchange

-C-

C^2	Command and Control
C^4I	Command, Control, Communication, Computers, and Intelligence.
CGF	Computer Generated Forces
COEA	Cost and Operational Effectiveness Analysis

-D-

DIS	Distributed Interactive Simulation
DoD	Department of Defense
DOF	Degrees of Freedom

-H-

HLA	High Level Architecture

-J-

JointSAF	Joint Semi-Automated Forces

-M-

M&S	Modeling and Simulation
MOE	Measure of Effectiveness

-P-

PDU Protocol Data Unit

-S-

SBA Simulation Based Acquisition

SDG Supplemental Data Generator

References

[1] Johnson, V. R., McKeon, M. F., Szanto, T. R., "Simulation Based Acquisition: A New Approach," Defense Systems Management College, Report of the Military Research Fellows DSMC 1997-1998.

[2] George, Gary; Pope, Marie; and Mallery, Ellen; Hughes Training Inc., "Intelligent Agents for Computer Generated Forces", Prepared form Human Resources Directorate, Aircrew Training Research Division, Armstrong Laboratories, Mesa, Arizona 85206-0904, September 1996.

[3] Hanford, Ltc Craig B. PM, Air Combat Training Systems, STRICOM, and Knight, Samuel; Osgood, Micheal; Pope, Marie; Sardella, Joseph; Schwalm, Steve; Hughes Training Inc., Link Operations, "CELLNET; The Enhancement of Fielded Army Aviation Simulators to Provide DIS Interoperability and Constructive Simulation Linkage", 12th DIS Workshop on Standards for the Interoperability of Distributed Simulations, Volume I, Position Papers.

[4] Pope, Marie, "A CGF to meet the existing and future training requirements of the B-2 Aircrew Training Device", Proceedings of the 8th Conference on Computer Generated Forces and Behavior Representation, Orlando, FL 1999.

[5] Pope, M., & Miller, D.M., "A Legacy Model for Tomorrow's Training", Proceedings of the 17th Interservice/Industry Training Systems and Education Conference, Albuquerque, NM 1995.

[6] Simulation Technology Branch, System Concepts and Simulation Division, Avionics Directorate, Wright Laboratory, Air Force Material Command, Wright Patterson Air Force Base, OH, 45433-7334, *Suppressor Version 5.5 Documentation*, 1999.

[7] Ewen, D., Dion, D., Flynn, T., Miller, D., "Computer Generated Forces Applications to a simulation Based Acquisition Smart Product Model for SC-21"" Proceedings of the Ninth Conference on Computer Generated Forces, Orlando, Fla., 2000.