Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

MODULAR SEMI-AUTOMATED FORCES

Andy Ceranowicz

Loral Advanced Distributed Simulation
50 Moulton Street
Cambridge, Massachusetts 02138, U.S.A.

ABSTRACT

ModSAF or Modular Semi-Automated Forces is a combat simulation used to support Distributed Interactive Simulation (DIS). An important factor in creating realistic and valid virtual battlefields is the presence of many different battlefield entities including opposing and supporting forces. Despite the decreasing costs of simulators, the cost of fully populating the virtual battlefield with crewed vehicle simulators is prohibitive. Semi-Automated Forces (SAF) systems are virtual simulations that solve this problem by simulating multiple vehicles with a single operator in supervisory control. ModSAF is a third generation SAF system designed to make it possible for a single operator to control larger units and make it easier to add new battlefield entities to the simulation. The battlefield contains many significant objects which exhibit a wide range of behaviors and interactions. To capture these behaviors in simulation will take a broad multidiciplinary effort. ModSAF was designed with a modular open architecture so it could provide a platform for SAF and DIS researchers to build upon. ModSAF is a jointly sponsored research project of STRICOM and ARPA.

1 DISTRIBUTED INTERACTIVE SIMULATION

In DIS teams of soldiers in simulators interact with each other in virtual battles that they experience through computer generated sensory displays. These displays include out-the-window visuals produced by computer image generators. The potential benefits of applying DIS to many DoD activities have been widely recognized. The virtual battlefield provides an environment where soldiers can participate in exercises without the constraints of safety, cost, and availability associated with field exercises. These DoD activities, spanning the acquisition and operational life cycle, include team and command training, combat and materiel development, and weapons systems evaluation and acquisition. Modular Semi-Automated Forces (ModSAF) is a DIS system for simulating and controlling entities, such as vehicles, Dismounted Infantry (DI), missiles, and dynamic structures to populate the virtual battlefield.

2 SEMI-AUTOMATED FORCES

Realism in the DIS environment can be measured along a number of dimensions (McBride 1992) One of these dimensions is display quality. How realistic are the images through which the soldier perceives the virtual battlefield? A second dimension is behavioral quality. Is the behavior of the entities in the simulation realistic? Yet another dimension is the completeness of the environment. Are all the significant entities and phenomena represented in the correct numbers?

Distributed Interactive Simulation is made possible by the availability of low cost crewed simulators. This has allowed sufficient numbers of simulators to be available for up to battalion sized maneuver units to train together. However, acquiring and manning enough simulators to represent all the significant battlefield entities in sufficient numbers is economically prohibitive. In addition to the unit being trained or studied, the battlefield contains opposing forces (typically more numerous than the trainees), flanking forces fighting side by side with the training unit, support forces (providing fire support, supplies, and reinforcements), and noncombatants. The only economically viable way of populating the battlefield with all the types and numbers of interactive agents required for realism is to generate them by computer.

Because artificial intelligence has not yet provided us with intelligent computer agents, practical computer generated forces systems are either semi-automated, where an operator is in supervisory control of the agents, or constructive, where behavior is limited to narrow excursions from a carefully planned scenario. Work on the initial DIS system, SIMNET (SIMulator

NETworking), was started by ARPA in 1983 (Alluisi 1991). Semi-Automated Forces (SAF) have been a part of SIMNET since the fall of 1987 (Ceranowicz, Downes-Martin, Saffi 1988, 1989). A Semi-Automated approach was chosen over a constructive one because the restriction to carefully planned scenarios was too limiting to achieve the kind of free-play man-in-the-loop environment ARPA was aiming for. Constructive simulations, or computerized wargames, are continuously evolving. As they take advantage of the increased computational power and advanced user interface features offered by workstation technology, wargames are evolving to support more open scenarios. In fact, Janus is one constructive wargame that is being extended to interface directly with DIS simulations. However, constructive wargames have made extensive use of statistical models to deal with many different phenomena efficiently. The interoperability of these statistical models with the explicit visual models of DIS is problematic. For example, intervisibility may be based on an average density of vegetation in a wargame and on explicitly represented trees and shrubs in a simulator. Interoperability is made more difficult by the DIS capability for After Action Review. DIS exercises are recorded and after each exercise troops review it to study their mistakes. A typical question is "where was the enemy unit that killed me?". If the playback shows that the firer could not have shot/detected the crewed simulator under the explicit model of the situation, then the effectiveness of the entire exercise is compromised.

Since its SIMNET beginning, the goals of SAF research have remained the same: provide larger numbers and varieties of increasingly realistic agents, operating in increasingly realistic environments, for lower cost. These goals are a challenge to the technology available to us today. Increasing the efficiency of simulating interactive agents implies reducing the human and/or the computational resources required. Reducing the amount of human supervision requires more efficient user interfaces and more automated agents which in turn require more computational resources. We have focused on reducing the human resources. Unfortunately, the requirements for improved behavior, more accurate physical models (e.g., higher resolution terrain), and the inclusion of more battlefield phenomena (e.g., smoke, flares) are increasing the computational requirements much faster than the increases being provided by advances in computer technology. We need to find more efficient simulation algorithms to support the needs of advanced distributed simulation. Simulation of a wide variety of agents and organizations also requires large amounts of domain knowledge organized into systems capable of producing intelligent behavior in real time. This knowledge will constantly require updates as DIS technology matures and real world equipment, tactics, and crisis locations change. Semi-Automated Forces is a problem whose solution will require the combined efforts of many researchers and organizations.

3 MODSAF OVERVIEW

ModSAF is the third generation of SAF systems developed at Loral Advanced Distributed Simulation (formerly part of Bolt, Beranek, and Newman). Its goal is to advance SAF technology by providing an architecture that will scale up to allow operators to control higher echelon units and allow the addition of a wide variety of battlefield entities. In recognition of the magnitude of the SAF problem, ModSAF has a modular open architecture to enable other researchers to build on it and focus on more difficult problems.

The ModSAF system includes three components: the SAFstation, the SAFsim, and the SAF-logger. The SAFstation (SAF workstation) provides the graphical user interface from which the operator initializes exercises, observes the battle, and commands his semi-automated forces. The SAFsim (SAF simulator) simulates all the SAF entities, units, and environmental processes. The SAF-logger saves exercises by recording the network traffic including DIS packets and behavioral information from ModSAF. These ModSAF components are typically run on separate computers connected by a network. For small experiments and development the SAFstation and the SAFsim can be run on one computer to reduce the amount of hardware required.

The SAFstation allows a user to monitor and control ModSAF forces and setup exercises. The SAFstation has an electronic map and situation display which allows the operator to view the terrain and monitor the positions and actions of his forces and the enemy forces they are in contact with. Terrain analysis tools provide the ability to evaluate visibility and terrain cross-sections. Drawing tools are used to mark the map with control measures such as phase lines, control points, and battle positions. The operator creates missions by filling in an execution matrix. The execution matrix allows the operator to divide the mission into a sequence of phases. For each phase he specifies the tasks each subordinate unit is to execute and the transition criteria to the next phase. Custom task editors for each task allow the operator to specialize tasks for a particular mission. Control measures from the map are available as terrain references for specifying phase transitions (start assault after all units reach phase line Charlie) and as arguments to tasks (assault objective Dog). The operator can issue missions to units and modify them during execution. Missions can be interrupted or have their parameters, such as objectives or routes, modified by issuing immediate commands. A special facility exists to accelerate the issuing of immediate commands to allow the operator to react quickly new situations. A message log records reports from the forces commanded by the workstation and a situation monitor allows the operator to monitor the state of their tasks. Because the role of the operator is to augment the automated logic embedded in the SAF entities to produce plausible behavior (Ceranowicz et. al. 1994) he is allowed to know everything his units know and to issue direct orders to any of the units or subordinates under his control. While this may seem like an unfair advantage, in practice, the operator's work load is such that he can only afford to pay attention to a small subset of the information available to him just like a normal commander. Until SAF agents are intelligent enough to respond to commands in the same way as human agents, the SAF operator will need access to more information about the units than would be available in the real world. The SAFstation also allows the operator to control the positioning of a stealth out-the-window-view which allows him to view his forces in 3-D perspective. This is extremely useful for inspecting and adjusting vehicle positions for ambushes and battle positions. In addition to these command interfaces, the SAFstation has a password protected battlemaster mode that allows the operator the set up exercises by creating and placing the order of battle and saving and restoring scenarios. The SAFstation does no simulation, all simulation is done by one or more SAFsims. This division of labor is advantageous, allowing a variety of systems to generate missions for a ModSAF unit, including different workstations, AI programs, as well as other SAFsims.

The SAFsim component is a real time time-stepped simulation. ModSAF does not enforce a constant update interval but instead allows it to float with simulation load. This allows ModSAF to run with standard operating systems where disk activity or other operating system events can occasionally cause long ticks. It also allows performance to degrade gracefully as simulation load is increased. Under full load it is designed to operate at a 2 Hz update rate. It maintains an object based database of the state of the battlefield. This database contains two types of objects: local entities whose state is updated by the SAFsim and remote entities whose state is updated by other SAFsims or simulators on the network. The state of remote entities is updated from entity state packets received from the network. For local entities, a scheduler periodically runs their simulation models and updates their state. As the internal state gets updated, entity state packets are transmitted to inform other simulators of their state. Events such as weapons fire, munition impacts, and radio transmissions are also sent out as packets to which the local vehicles compute responses. This process of maintaining an internal database of the state of entities simulated by other computers is central to distributed simulation. The entity state information sent out is always ground truth. It is up to the simulator of each vehicle to determine which other vehicles it should know about.

Simulation of entities in ModSAF is divided into the behavioral and physical simulation. ModSAF physical models are composed by combining components such as hulls, turrets, weapons, and sensors. Updating the physical model of an entity is just the process of updating each of its components. The simulation of behavior in ModSAF is done by tasks. Each entity has its own scheduler, called a task manager, that collects all the entity's tasks, orders them, and then executes them. Tasks are used to scan sensors, acquire targets, control weapons, control movement, and coordinate the activities of subordinates. Tasks and physical models interact through standardized interfaces called generic model interfaces. The organization of entities into units is explicitly represented by a network of "unit" objects. The way in which these objects are linked defines the organization of a unit and can be changed dynamically. The behavior of a unit is defined by the unit's mission and tasks. All of a unit's tasks are executed by the entity that represents the leader of that unit.

The SAF-logger records and plays back the events that occur during an exercise. The SAF-logger reduces storage requirements when recording by compressing the entity state information and saving only the changes in the packets. Periodically the entire entity state is saved (typically once a minute) so that you can seek to points in the recorded exercise without having to go back to the beginning of the exercise to reconstruct the state. Only those packets that are expected to have a high correlation such as entity state packets are compressed. The SAF-logger supports VCR-like special effects including freeze frame, fast forward, slow motion, and reverse. The SAF logger records both the physical state of the ModSAF units and the behavioral state. From this, the SAF logger can create scenario files that can reinitialize the ModSAF forces to any point in a recorded exercise and let them reexecute the remainder of the exercise.

4 SCALING TO HIGHER ECHELON CONTROL

ModSAF is a fully distributed system that allows unit simulation to be distributed over many computers while allowing one operator to control the entire unit.

SAFstations and SAFsims running distributed over a network communicate physical battlefield state and events between themselves via the DIS protocol and command, control, and system information via the Persistent Object (PO) Protocol (Calder et. al. 1993). Database abstractions are used to simplify these interfaces and make them uniform for both remote and local components. The PO Database allows distributed SAFsims and SAFstations to share information about the behavioral state of entities in the virtual battle. This behavioral state includes the entities' missions and their status, current tasks, and world views. An exercise using ModSAF can be configured in a variety of ways making optimal use of the hardware for that application. The SAFsims work together to provide a simulation server that automatically partitions the simulation load between the individual SAFsims. For exercises requiring large numbers of entities you would allocate more computers to run as SAFsims increasing your simulation resources. One user at one SAFstation can control a hierarchy of SAFsims to simulate large units with many entities. For exercises that require low level human control of SAF entities, you can allocate more computers to be SAFstations. A single SAFsim can support multiple SAFstations.

The fully distributed nature of ModSAF makes it possible for an operator to give commands to large number of vehicles. In order to effectively control those vehicles the operator's span of control must be increased. To support the operator, ModSAF provides a hierarchical behavioral representation that allows tasks executing at one level to spawn tasks and task frames on the lower levels. Thus when an operator gives a company a mission, the company mission spawns task frames for its platoons to execute, and the platoon task frames spawn tasks for the vehicles to execute. This hierarchical behavioral representation allows the operator to give tasks at higher levels and should be able to scale well above company level. Heavy use is made of default and computed parameters in these missions to reduce the number of parameters that the user has to specify. For example, in the process of occupying a battle position, previous SAF systems had no capability of selecting covered and concealed positions. These had to be supplied by the user. ModSAF now computes these positions for the user (Longtin 1994).. If the operator is not satisfied with the way that his units are executing their missions a special immediate command facility is provided to allow the operator to modify their behavior. Many mission parameters are available for direct manipulation on the electronic map allowing the operator to change unit missions by simply changing the control measures on the map. These capabilities are reducing the amount of effort it requires to control ModSAF entities allowing the operator to control larger units.

The operator's role in a semi-automated forces system includes intervening to handle those situations where the automated decision logic in the entities is not sophisticated enough to react properly. The less robust the behavior of the SAF entities is, the more intervention the operator must perform and the fewer entities he can control. ModSAF provides many behavioral utilities that support the generation of more robust entity behavior. The avoidance of obstacles is one area that has traditionally been difficult for SAF systems. At the individual entity level ModSAF uses a short term planner to allow entities to navigate around multiple simultaneous obstacles (Smith 1994). At the unit level ModSAF uses a route planner to avoid larger obstacles and to determine where the unit needs to change formation. Expanded architectural support for reactions makes it easier for the operator to control reactions. At any time the operator can stop a reaction. The reaction will automatically become active again once the trigger situation for the stopped reaction goes away, so that the system will react to future trigger situations. If the operator wants to permanently alter the reaction he can do so by modifying the reaction parameters at any time.

5 SOFTWARE ARCHITECTURE

ModSAF provides this simulation capability in a modular software architecture that encourages users to extend and modify the system to support their applications. Over 99% of ModSAF software is implemented as library modules with strictly defined and documented public interfaces. Each library provides a service, such as vector math utilities, network interface, tactical map drawing, physical module simulation, or behavior simulation. Layering and callback techniques are used to minimize module interdependence. Small main programs link together these libraries to form ModSAF applications making it easy to build custom applications. The concept underlying this software organization is that these libraries form a repository of useful capabilities that can be combined in different ways to produce a variety of DIS and SAF systems. Researchers can add new libraries and applications to this repository or make use of existing libraries in their own systems. Higher level command systems can also be built on top of ModSAF to control ModSAF entities.

The ModSAF architecture is object-based, dividing the world into distinct objects whose activities are simulated individually. To make it easier to add new entities to ModSAF, the definition of ModSAF simulation entities and units is done at program start up by reading parameter files. The parameter files not only determine what the parameters of an entity model are going to be, they actually determine the software modules that will be used to simulate the entity. These entity definitions can be changed dynamically during runtime via the configuration editors that specify new object definitions via the PO Database. Because entity definitions are data driven, the best way to describe the capabilities of ModSAF is by describing the models that can be linked together to form entities. Some of these models can be instantiated multiple times per entity and some can only be instantiated once. Those models that have generic model interfaces defined for them can be interchanged and be controlled by the same tasks. Examples of these models include:

• Dynamics Models: These models are controlled through the Hulls generic model interface. They include tracked, wheeled, rotary wing, and fixed wing vehicle models. Missiles and infantry models are also available. Only one instantiation is allowed at a time for each entity.

• Turret Models: These models are controlled through the Turrets generic model interface. Multiple instantiations per entity are allowed. Currently there is only one turret model. It can be configured with weapons and sensor models to allow them to change orientation.

• Weapons Models: These models are controlled through the Guns generic model interface. Multiple instantiations are allowed. These models include a probabilistic hit model used for cannons and machine guns and a missile launcher model that is used to create and launch missile entities that then fly out to the target.

• Sensor Models: These models are controlled through the Sensors generic model interface. Multiple instantiations are allowed. Models include radar, visual, and IR.

• Damage models implement damage assessment from indirect fire and direct fire weapons. These models support catastrophic, firepower, and mobility kills.

These component models have been used to implement physical models of the following vehicles.

FWA-	F14D, MIG 29, A10, SU25
RWA -	AH64, 0H58D, Mi24, Mi28
Tanks -	M1, T72, T80, Leo2, Leo1A5,
Mechs -	M2, BMP1, BMP2
Howitzers -	M109, 2S1
Mortars -	M106
ADA -	ZSU-23/4
DI -	Various group DI units
CSS -	M977
Other-	M113 Ambulance

Behavior in ModSAF is implemented by tasks. Vehicle (or individual) tasks provide basic behavior for single vehicles. Examples of the individual vehicle tasks implemented in ModSAF include:

• Move generates commands to its hull to move the vehicle along its route and around obstacles. Move can selectively avoid rivers, lakes, buildings, trees, tree lines, and other vehicles.

• Terrain maintains a map of local obstacles for the vehicle to avoid.

• Search slews the vehicle's sensors to scan its surroundings.

• Targeter controls the vehicle's weapons to fire on a target. It may also control vehicle motion to obtain a firing solution.

Unit tasks represent independent actions by a unit or individual entities. Unit tasks are executed by the leader of a unit. The following are examples of unit tasks that have been implemented in ModSAF:

• Assault issues movement and firing commands to perform an attack and secure the position attacked.

• Actions On Contact determines how to react to enemy contact. Its decision criteria include the number of enemy vehicles and whether they are attacking.

• Traveling moves unit in formation or performs roadmarch. Formations include line, column, vee, wedge, echelon, right/left, and staggered column.

These tasks can be combined together to create sets of commands called "task frames" that define what a unit does during a mission phase.

6 PROGRAM STATUS

The initial version of ModSAF was released in May 1993. It supported beyond-visual-range air-to-air combat simulation. In December 1993, ModSAF 1.0 was released. It extended ModSAF capabilities to include the simulation of platoon level tank and mechanized infantry forces as well as a number of individual vehicle simulations. ModSAF 1.0 contained 150 libraries, 215K lines of source code, 67K lines of code comments, 27K lines of data files, and 103K lines of online documentation. ModSAF 1.2 was released in July 1993. ModSAF 1.2 extended ModSAF to include company level maneuver forces with a limited dismounted infantry capability, flights of rotary wing aircraft, close air support in flights, a fire support model including howitzers and mortars, minefields and breaching, air defense artillery platoons, and the provisioning of fuel and ammunition from combat service support vehicles. ModSAF 1.2 contained 255 libraries, 394K lines of source code, 72K of data files, and 141K of online documentation. Future releases of ModSAF are scheduled quarterly and will include development from other projects that will be merged into ModSAF.

Future ModSAF development will be driven by the requirements of the STRICOM and ARPA programs that it is supporting. Major programs driving ModSAF requirements include A2ATD (Anti-Armor Advanced Technology Demonstration) and STOW (Synthetic Theater Of War). A2ATD is intended to develop and demonstrate a verified, validated, and accredited (VV&A) DIS testbed capability to support combat and materiel development studies (Courtemanche, Monday 1994). This will result in the VV&A of ModSAF. STOW has the objective of demonstrating the use of DIS for large scale exercises distributed over many sites. In this effort ModSAF is being linked to the BBS wargame.

7 CONCLUSION

ModSAF or Modular Semi-Automated Forces is the successor to the ARPA SIMNET and ODIN Semi-Automated Forces systems. It provides SAF simulation capabilities in an open, modular architecture with full government rights, that DIS and SAF researchers can build upon and extend. It contains no COTS (Commercial Off the Shelf) or proprietary components and runs on a variety of UNIX platforms. Its architecture is designed to allow it to scale up to the simulation of larger units while increasing simulation realism. ModSAF is available from the Tactical Warfare Simulation and Technology Information Analysis Center at the University of Central Florida.

ACKNOWLEDGEMENTS

To provide an environment in which many SAF researchers from different organizations could combine their efforts, CDR Dennis K. McBride of ARPA, started the development of Modular Semi-Automated Forces under the WISSARD program in 1992. The ARMY has jointed in support of this concept, with Stan Goodman from STRICOM, taking charge of basic ModSAF development through the Advanced Distributed Simulation Technology program. This work is being supported by the Ground ModSAF project, contract number N66001-92-D-0058, and the ADST program, contract number N61339-91-D-0001-0021.

REFERENCES

McBride, D. K., 1992. Simulation and High Fidelity on a Network. The 1992 L.P. Coombes Memorial Lecture, Melbourne, Australia.

Alluisi, E., 1991. The development of Technology for Collective Training: SIMNET, a Case History. Human Factors 33(3), 343-362.

Ceranowicz, A. Z., Downes-Martin, S., Saffi, M., 1988. SIMNET Semi-Automated Forces Version 3.0: Implementation Issues. BBN Systems and Technologies Corporation Report 6940.

Ceranowicz, A. Z., Downes-Martin, S., Saffi, M., 1989. SIMNET Semi-Automated Forces Version 3.0: A Functional Description. BBN Systems and Technologies Corporation Report 6939.

Ceranowicz, A., Coffin, D., Smith, J., Gonzalez, R., Ladd, C., 1994. Operator Control of Behavior in ModSAF. Fourth Conference on Computer Generated Forces and Behavioral Representation, University of Central Florida, Orlando.

Calder, R. B., Smith, J. E., Courtemanche, A. J., Mar, M. F., Ceranowicz, A. Z., 1993. ModSAF Behavioral Simulation and Control. Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation., University of Central Florida, Orlando.

Longtin, M., 1994. Cover and Concealment in ModSAF. Fourth Conference on Computer Generated Forces and Behavioral Representation, University of Central Florida, Orlando.

Smith, J. E., 1994. Near-term Movement Control in ModSAF. Fourth Conference on Computer Generated Forces and Behavioral Representation, University of Central Florida, Orlando.

Courtemanche, A. J., Monday, P., 1994. The Incorporation of Validated Combat Models into ModSAF. Fourth Conference on Computer

Generated Forces and Behavioral Representation, University of Central Florida, Orlando.

AUTHOR BIOGRAPHY

ANDY CERANOWICZ is the manager of the Semi-Automated Forces group at Loral Advanced Distributed Simulation. He has been working on Distributed Interactive Simulation and Semi-Automated Forces since 1986 when he joined BBN to work on the ARPA Simulation Networking (SIMNET) Project. His research interests lie in the applications of distributed simulation and the design of intelligent interactive agents. He is a member of the IEEE.