Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

AN EXPERIMENT IN SIMULATION INTEROPERABILITY

Gary N. Bundy
David W. Seidel
Ben C. King
Carl D. Burke

The MITRE Corporation
1820 Dolley Madison Blvd
McLean, Virginia 22102

ABSTRACT

The AWSIM Interoperability with ModSAF (AIM) program is investigating interfacing traditional constructive simulations with virtual simulations. The program addresses a number of issues involved in this interoperability. As a mechanism for exploring these issues, ModSAF is used to create a detailed window into an AWSIM exercise; modeling of AWSIM aircraft entering the window is transferred to ModSAF. A prototype was developed to investigate the practical aspects of the interface issues. Knowledge, algorithms, and software developed in this program will support future Air Force wargame development.

1 INTRODUCTION

The AWSIM Interoperability with ModSAF (AIM) program is investigating approaches and issues in interfacing traditional constructive simulations with virtual simulations.

The form of integration assumes a broad constructive simulation within which a virtual simulation portrays a detailed window (virtual playbox). In this case, the constructive simulation is AWSIM, an Air Force simulation widely used in theater-wide computer-aided exercises; and the virtual simulation is ModSAF, a detailed simulation used in conjunction with vehicle trainers. The mechanism for exchange is the Aggregate Level Simulation Protocol (ALSP), a mechanism used in theater training exercises.

When aircraft modeled in AWSIM cross into a ModSAF window, responsibility for modeling the aircraft is transferred to ModSAF. When the aircraft cross out of the window, modeling responsibility is passed back to AWSIM. AWSIM retains command and control responsibility for the aircraft regardless of where it is modeled.

1.1 Issues

Issues that form the framework for the AIM investigation include the following:

- **Deaggregation/Reaggregation**. When an aggregate unit in a constructive model is decomposed into its constituent entities, the entities should be properly placed on the battlefield with sufficient information to continue the mission of the unit when responsibility for modeling the entities is transferred to the virtual simulation. Similarly, when a unit reaggregates, the constructive simulation should receive sufficient information about the unit to reflect the results of battlefield activities that occurred in the virtual simulation.
- **Coherency**. Consistent views of the battlefield should be maintained in both constructive and virtual simulations, including the state of the entities on the battlefield, the behaviors and activities that they perform, the time being represented, and the terrain that forms the context of the battle. A seamless transition of modeling and interactions should be present across the boundary between the simulations.
- **Command and Control**. Units in both the constructive and the virtual simulations should be able to accept command and control directives in a consistent way.
- **Exercise Support**. Mechanisms should be provided to facilitate the management of the total exercise. This entails an understanding of the total state of the exercise and coordinated control of the component pieces.

1.2 Constraints

Several constraints drive the AIM project:

- **Practicality**. The results of the investigation should be valid in the world of military exercises and training.

  AWSIM and ModSAF are simulations that are in regular use.

  Entities used in the prototype are real-world aircraft types, missions, and environment.

- **Reuse**. Wherever possible, existing software and processes should be used. In addition, where future related software development is contemplated, software and processes developed under this program should be transferable to the new program.

  The Aggregate Level Simulation Protocol (ALSP) is an existing information-exchange mechanism. AWSIM was already adapted to use ALSP and ALSP already provides mechanisms for ground-truth data exchanges and time management.

- **Visibility**. The exchange of information between the simulations should be explicit and externally visible.

  Information is exchanged using a message-based medium.

- **Maintaining Model Credibility**. In order to maintain the credibility of each simulation, new modeling should be avoided. Similarly, new software elements that introduce modeling concepts should be avoided.

1.3 Implementation Approach

The approach chosen to implement AIM was to use an enhanced ALSP protocol (See Figure 1). Since AWSIM was already adapted to exchange information and regulate time using ALSP, much of the effort in adapting AWSIM was already accomplished. What remained was

- Adapt ModSAF so that it performs basic ALSP interface functionality (broadcast modeled aircraft characteristics, receive and portray aircraft modeled in AWSIM, and coordinate time via ALSP).
- Adapt ModSAF to perform AIM-specific functionality (coordinate hand-off from and to AWSIM via ALSP, process and generate weapons interactions, and coordinate command and control from AWSIM via ALSP).
- Adapt AWSIM to perform AIM-specific functionality (coordinate hand-off from and to ModSAF via ALSP and coordinate command and control to ModSAF aircraft via ALSP).
- Adapt the ALSP protocol to handle the new requirements presented by AIM (hand-off and command and control).

2 SIMULATION COMPARISON

AWSIM and ModSAF differ in how they perceive and model air warfare. This section describes elements that are relevant to bringing them together in AIM. Paragraphs below depict how objects are described in each of the simulations and how each simulation models relevant activities.

2.1 Object Description

In AWSIM and ModSAF, objects are represented using attributes. This representation comes into play when one simulation must internalize (ghost) an object modeled by the other and when modeling responsibility must pass from one simulation to the other. Table 1 compares significant data elements as they are represented in the two simulations

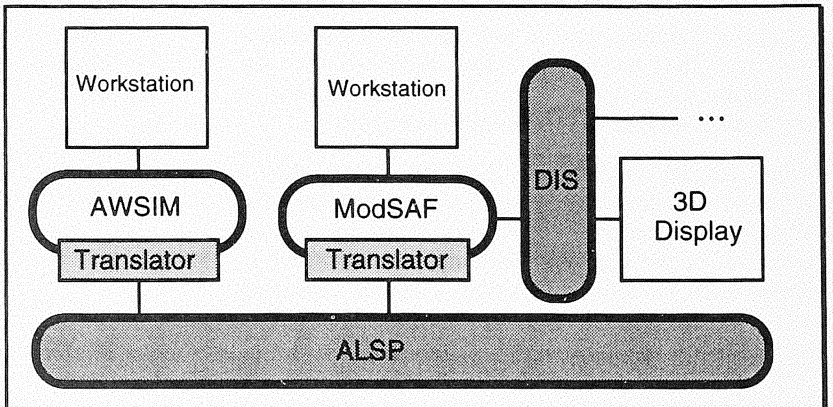

Figure 1. AIM Architecture

Table 1. Object Description Comparison

Data type	ModSAF	AWSIM
Entity represented	Individual aircraft	Flight of aircraft
Flight name	None	Used by AWSIM to track and control aircraft.
Location	3-dimensional distance from the center of the earth	Latitude, longitude, altitude above mean sea level
Attitude	Attitude of each aircraft.	Not modeled
Dead Reckoning parameters	Rate of change of positional and attitude data.	Course and speed.
Tail number	Marking text represents tail number	For reporting purposes only.
Formation	Formations can be specified	Not modeled
Sensor status	Most sensors are not modeled	Modeled
Weapon load	Quantity of each type of weapon on each aircraft	Aggregate quantity of each type of weapon for the flight.
Fuel	Maintained for each aircraft	An average fuel load for all aircraft in the flight is kept

2.2 Modeling

In AWSIM and ModSAF, activities of single entities and interactions between entities are modeled in different ways. These differences are significant when comparing similar activities that take place separately in each simulation (results of AWSIM air-to-ground attack versus results of ModSAF air-to-ground attack) and when attempting to integrate activities where participating objects are modeled in different simulations (air-to-air engagement between AWSIM and ModSAF aircraft). Table 2 compares activities as they are modeled in the two simulations.

3 RESOLUTION APPROACHES

The differences between the simulations that are described in Section 2 must be rectified if an interoperable federation of simulations is to exist. The most obvious and straightforward approach to accomplishing this is to add representations and modeling to each simulation to make it match the other. However, this is not the best approach for a number of reasons:

- Some abstractions are purposely built into a simulation (1) to permit its users to deal with the activity being modeled at a higher level (2) to ease scenario development or (3) to ease processing requirements, permitting large scenarios to be executed in relatively small computers.
- The cost to add detail to the more abstract simulation is excessive.
- Implementation of detailed modeling is often not required since what is evident is appearance at the point of interface.

Therefore, three approaches to the rectification present themselves:

- **Add modeling**. New modeling of activity can be added to a simulation. For example, in AWSIM, fuel could be modeled (consumed, replenished when refueled) for each aircraft in a flight individually. As discussed above, adding the modeling of a specific function is the most costly alternative.
- **Add representation**. New data elements can be added to a simulation. For example, in AWSIM, a flight formation could be kept. While this approach may be adequate, it often introduces simplistic, operationally inappropriate results.

Table 2. Battlefield Activity Comparison

Activity	ModSAF	AWSIM
Entity Update rate	Often enough that, using dead reckoning parameters, another simulation could calculate location and attitude within prescribed values (typically about once per second)	Selectable (once every 10 seconds to once per minute).
Ground Entities	ModSAF models ground vehicles • Permits better target selection for air-to-ground attacks. • Permits more detailed flight profiles • Permits more rapid reattack decisions	AWSIM models only SAMs, airbases, and search radars. In ALSP exercises, AWSIM passes target preferences in ALSP air-to-ground interactions; the ground simulation (CBS) selects specific targets.
Jamming	Not modeled	AWSIM models self protection and stand-off jamming
Terrain	Detailed terrain representation is part of ModSAF. Outside the terrain box, no earth representation exists. Current air implementations do not use terrain.	AWSIM does not model terrain • No map-of-the-earth flying • No terrain masking of radars and ground-based weapons • No terrain effect on detection of ground targets in complex terrain
Air picture	ModSAF has a complete picture only of its playbox.	AWSIM keeps a picture of the total battlefield through ALSP.
Command and Control	Direct from GUI	Future tasking can be input (ATO-like orders)
Time	ModSAF and DIS evolve time in very small "ticks" and lock time advance to the wall clock.	AWSIM advances time in fixed (10 second to 1 minute) increments. It attempts to synchronize simulation time to wall clock time but may fall behind if processing needs are heavy. In ALSP confederations, AWSIM regulates its time through ALSP.

- **Create information**. New information can be introduced at the time of exchange of data. For example, AWSIM could apply some algorithm to determine individual aircraft fuel state and broadcast that information. Alternatively, AWSIM could broadcast flight fuel state and ModSAF could apply an algorithm to determine individual aircraft fuel state. In general, it appears best to assign this role to the simulation with the most detailed modeling of the particular function—ModSAF in this case.

It must be noted that all of these approaches create information that is not present when the simulations operate separately. By doing so, any validation or accreditation of either of the simulations is abrogated; the process must be performed again against the simulations interoperating together.

4 AIM IMPLEMENTATION METHODOLOGY

Given the differences in representation and modeling described above, mechanisms were chosen to permit interoperability between AWSIM and ModSAF. Each of the mechanisms entail compromises between the desired seamless integration and the practical requirement to integrate without violent disruption to either simulation.

This section describes the compromises that have been made and resulting modeling discrepancies. The section is divided according to the interoperability issues described in Section 1.

Table 3. Deaggregation/Reaggregation Issues

Issue	Resolution	Resulting Anomalies
AWSIM doesn't model attitude	ModSAF creates attitude data for each aircraft	Aircraft attitude is jerky and unrealistic.
AWSIM doesn't model formation	ModSAF creates formation when it initially observes the flight.	Formations may be inappropriate. A rapid deaggregation, reaggregation, deaggregation sequence may lead to inconsistent formations.
ModSAF doesn't model Flights of aircraft	AWSIM passes flight data in ALSP message. ModSAF remembers and uses this data when reporting on the flight.	None
Units of measure differ between ModSAF and AWSIM	ALSP units of measure are used for information exchange.	None
AWSIM aggregates weapon and fuel loads	Each time a flight of aircraft newly appears in ModSAF, ModSAF splits loads onto aircraft.	A deaggregation, reaggregation, deaggregation sequence may lead to inconsistent loads.

4.1 Deaggregation/Reaggregation

When a flight of AWSIM aircraft enters ModSAF's window of interest, ModSAF begins to ghost the flight. When the flight enters ModSAF's window of control, modeling responsibility is handed-off to ModSAF—ModSAF must broadcast information about the flight over ALSP. When the flight of aircraft leaves the window of control, the hand-off mechanism is invoked to transfer modeling responsibility for the flight back to AWSIM. These transfers require private information that is not normally carried in ALSP update messages; the private data was incorporated into the update messages and the hand-off message exchange. These issues are described in Table 3.

4.2 Coherency

Consistent views of the battlefield should be maintained in both AWSIM and ModSAF, including the state of the entities on the battlefield, the activities that they are performing, the time being represented, and the terrain that forms the context of the battle. A seamless transition of modeling and interaction should be present across the window boundary. These issues are discussed in Table 4.

4.3 Command and Control

Units in both the constructive and the virtual simulations should be able to accept command and control directives in a consistent way. These issues are discussed in Table 5.

4.4 Exercise Support

Mechanisms should be provided to facilitate the management of the total exercise. This is particularly true in exercises, like AIM, with multiple simulations and complicated topologies, These mechanisms permit an understanding of the total state of the exercise and coordinated control of the component simulations. These issues are described in Table 6.

5 CONCLUSIONS

While the AIM investigation and prototype implementation are incomplete, several conclusions can be drawn from progress to date.

- Complete, seamless interoperability is not practical, and, given the strengths of each simulation and the intended use of the federated system, not desirable.
- Reasonable work-arounds can be found for most simulation disconnects. Most unreasonable anomalies exist because fundamental modeling is lacking from a simulation.
- Exercise support is inadequate for this combination of simulations. ModSAF and DIS do not provide facilities to incorporate necessary features.
- Verification, Validation, and Accreditation (VV&A) of separate simulations does not produce a verified, validated, and accredited federation of simulations.

Table 4. Coherency Issues

Issue	Resolution	Resulting Anomalies
ModSAF has a limited geographic playbox.	ModSAF models aircraft in its window of control and ghosts those in its window of interest.	Aircraft outside the window of interest are invisible to ModSAF objects.
AWSIM updates location less frequently than ModSAF.	ModSAF dead reckons aircraft between AWSIM updates.	Kinematics of AWSIM aircraft appear jerky and unrealistic to ModSAF observers.
ModSAF models ground vehicles; AWSIM does not.	To attack ground targets, AWSIM passes target preferences in ALSP air-to-ground interactions for the ground simulation to use to determine specific targets.	ModSAF permits better target selection for air-to-ground attacks.
AWSIM models self protection and stand-off jamming; ModSAF does not.	AWSIM jamming is not turned on in AIM.	Electronic warfare is not a part of AIM.
AWSIM uses logical time; ModSAF time is tied to wall clock time	ModSAF is assigned the role of time controller; it broadcasts its perception of time and AWSIM adheres to it.	If AWSIM workload is heavy, it will fall behind wall clock time and lose synchronization with ModSAF
AWSIM assesses interactions less frequently than ModSAF.	ModSAF aircraft firing at AWSIM aircraft cause ALSP interaction messages to be sent to AWSIM. AWSIM determines the results of the interaction when its next increment of modeling occurs (up to one minute away). It then sends a revised ALSP update message showing the new state of the aircraft or its destruction.	The delay in ModSAF's learning the result of the weapon shoot could be interpreted by ModSAF as a miss and cause further weapon firings.
SAMs and Air-to-air missiles are not modeled as objects in AWSIM	When a ModSAF-controlled aircraft fires at an AWSIM-controlled aircraft, an ALSP interaction message is sent from ModSAF to AWSIM. ModSAF calculates the number of weapons that could have hit the target aircraft and includes that number in the interaction message. AWSIM processes the message and determines whether the weapons hit.	Since AWSIM aircraft cannot maneuver in the same manner as ModSAF aircraft in reaction to attack, new pH/pK values are needed for AWSIM; the source of this data is currently unknown
	When an AWSIM-controlled aircraft fires at a ModSAF-controlled aircraft, an ALSP interaction message is sent from AWSIM to ModSAF. AWSIM calculates the number of weapons that could have hit the target aircraft and includes that number in the interaction message. ModSAF processes the interaction message and determines whether the weapons hit.	Incompatible aircraft parameters preclude a good understanding of hit probabilities and interaction results.

Table 5. Command and Control Issues

Issue	Resolution	Resulting Anomalies
An operator must be able to realistically order flights of aircraft	AIM adaptations of the ALSP protocol permit the AWSIM operator to send command and control orders to ModSAF-modeled aircraft. These messages are limited to AWSIM-implemented mission assignments and weapon delivery. However, since AWSIM is an accepted Air Force simulation, the orders should closely adhere to Air Force doctrine	None
An operator must be able to receive reports from flights of aircraft	ModSAF generates status reports describing a flight of aircraft to AWSIM. The frequency and content of these messages are the subject of on-going knowledge acquisition; current implementation is an estimate of the result	None

Table 6. Exercise Support Issues

Issue	Resolution	Resulting Anomalies
It is useful to monitor the health of the simulations and their communications	ALSP infrastructure software provides this facility for the constructive confederation, and consequently also for AIM	None
A mechanism is needed to assure that only one simulation owns (controls the update of) a particular attribute of an entity or the entity as a whole	ALSP protocol and supporting infrastructure software provide this mechanism	None
During long exercises, it is often necessary to pause exercise activity in all simulations simultaneously. This requires automated control over all simulation clocks. AWSIM and ALSP provide this control; ModSAF and DIS do not	No automated pause capability is provided in AIM.	AIM exercises must proceed to completion without pause in simulation time.
During long exercises, it is often necessary to periodically store the current state of the simulations outside of simulation time (time stops while state save occurs). AWSIM and ALSP provide a mechanism for synchronizing the state saves so that all states represent the same instant in simulation time; ModSAF and DIS do not.	No automated state save capability is provided in AIM.	No facility is available to recover from simulation crashes or to reverse operational decisions during the exercise.

ACKNOWLEDGMENTS

The AIM project is sponsored by the US Air Force's Electronic Systems Center (ESC). It builds on previous efforts in support of the Synthetic Theater of War (STOW) program that was sponsored by the Defense Advanced Research Projects Agency (DARPA).

REFERENCES

Ceranowicz, A. Z, J. E. Smith, A. J. Courtemanche, R. B. Calder, November 1992, *ModSAF Programmer's Guide*, Loral Advanced Distributed Simulation, Inc., Cambridge MA.

Defense Modeling and Simulation Office, July 1993, *Survey of Semi-Automated Forces*, Arlington VA.

DIS Steering Committee, October 1993, *The DIS Vision, A Map to the Future of Distributed Simulation*, Orlando FL.

Institute for Simulation and Training, February 1994, *Standard for Distributed Interactive Simulation—Application Protocols, Version 2.0 Fourth Draft*, Orlando FL

Institute for Simulation and Training, June 1993, *Communication Architecture for Distributed Interactive Simulation (CADIS) Guidance Document*, Orlando FL

Institute for Simulation and Training, June 1993, *Communication Architecture for Distributed Interactive Simulation (CADIS) Proposed IEEE Final Draft Standard*, Orlando FL

Institute for Simulation and Training, June 1993, *Communication Architecture for Distributed Interactive Simulation (CADIS) Rationale*, Orlando FL

Loral Advanced Distributed Simulation, Inc., February 1993, *ModSAF Software Architecture Design and Overview Document*, Cambridge MA.

Miller, G. R., A. R. Adams, D. W. Seidel, November 1993, *Aggregate Level Simulation Protocol (ALSP) 1993 Confederation Annual Report*, MITRE Corporation, McLean VA.

MITRE Corporation, April 1996, *Aggregate Level Simulation Protocol (ALSP) Technical Specification,* The MITRE Corporation, McLean VA (not in public domain).

MITRE Corporation, February 1996, *Aggregate Level Simulation Protocol (ALSP) 1996 Joint Training Confederation Operational Specification,* The MITRE Corporation, McLean VA (not in public domain).

MITRE Corporation, April 1995, *Aggregate Level Simulation Protocol (ALSP) Infrastructure Software User Manual,* The MITRE Corporation, McLean VA (not in public domain).

MITRE Corporation, March 1996, *AWSIM - ModSAF Interface Control Document*, McLean VA (not in public domain).

AUTHOR'S BIOGRAPHIES

GARY N. BUNDY is the technical leader for the AIM effort at The MITRE Corporation. He has been with MITRE for over five years. During that time he has supported DoD agencies in integrating military simulations as well as integrating military simulations with C4I systems.

DAVID W. SEIDEL is project leader for the AIM effort at the MITRE Corporation. For the past five years he has supported DoD agencies in integrating military simulations. He has over fifteen years of experience in development and application of military wargames.

BEN C. KING has been supporting modeling and simulation interfacing at the MITRE Corporation for five years. In addition to leading the technical effort on AIM, he implemented key portions of ALSP and Command Forces software.

CARL D. BURKE is has performed software development and Army systems analysis at the MITRE Corporation for ten years. His previous modeling and simulation experience includes support for the JANUS interface to BDS-D.