Knowledge-Rich Intelligent Agents for SOF Training

Randolph M. Jones, Brian S. Stensrud, Glenn E. Taylor
Soar Technology, Inc.
Waterville, ME; Orlando, FL; Ann Arbor, MI
rjones@soartech.com, stensrud@soartech.com, glenn@soartech.com

ABSTRACT

This paper provides an overview of a variety of applications of knowledge-based systems and intelligent agents to modeling and simulation for special operations training. There are a number of ways in which intelligent agents can support training; including modeling special-operations individuals, other members of the blue forces, as well as opponent forces. The paper presents examples of each of these and provides a description of the knowledge acquisition and engineering efforts we have developed to build such systems. Applying intelligent agents to simulation-based training provides new opportunities to develop training exercises that are more consistent, less expensive, and more portable. These all provide opportunities for special operations members to train individually when they have time, rather than needing to schedule large-scale events.

ABOUT THE AUTHORS

Randolph M. Jones, Ph.D., is the scientific lead for Soar Technology's behavior modeling for kinetic simulations, as well as internal research and development. He has worked with a variety of agent architectures and models, including having participated in the TacAir-Soar project since its inception, where he wrote the first implementation of the TacAir-Soar system. He has over 20 years of experience doing research on agent architectures, intelligent agents, machine and human learning, graphical user interfaces, cognitive modeling, and a variety of related areas. Dr. Jones was an Assistant Professor and Chair of Computer Science at Colby College, and previously held research positions at the University of Michigan, the University of Pittsburgh, and Carnegie Mellon University. His general areas of research include architectures for knowledge-rich intelligent agents, computational models of human learning and problem solving, executable psychological models, computer games, automated intelligent actors, and improved usability and visualization in information systems. He earned a B.S. (1984) in Mathematics and Computer Science at UCLA, and M.S. (1987) and Ph.D. (1989) degrees from the Department of Information and Computer Science at the University of California, Irvine.

Brian S. Stensrud, Ph.D., is a Research Scientist at Soar Technology and lead behavior developer on the IF-Soar effort described in the paper. Brian has designed and developed a variety of intelligent agent systems during his tenure at Soar Technology, and has over seven years experience in the areas of knowledge-based systems and artificial intelligence. Dr. Stensrud received his Ph.D. in Computer Engineering from the University of Central Florida in 2005. He also holds B.S. degrees in Computer Engineering, Electrical Engineering, and Mathematics from the University of Florida.

Glenn Taylor is a Senior Scientist at Soar Technology. His R&D activities include interactive cognitive, social, and cultural human behavior models. He received his BS and MS in computer science and engineering from the University of Michigan, and has been at Soar Technology since 1998.

Knowledge-Rich Intelligent Agents for SOF Training

Randolph M. Jones, Brian S. Stensrud, Glenn Taylor
Soar Technology, Inc.
Waterville, ME; Orlando, FL; Ann Arbor, MI
rjones@soartech.com, stensrud@soartech.com, glenn@soartech.com

INTRODUCTION

There is increasing interest in providing improved access to effective military training through the use of computer simulations. This is becoming particularly important for training personnel in the use of tactical decision-making skills necessary for success in conventional, asymmetric, and non-kinetic warfare. When it comes to training, the US Department of Defense (DOD) faces the following challenges:

- There is a limited supply of qualified instructors. New instructors must be experts, who are expensive to train and retain.
- There are a decreasing number of opportunities to train.
- The current environment of continuous operations is stressing the effectiveness of training.
- Training costs continue to rise.

As a result, the number of "critical training experiences" that a warfighter can be exposed to is increasingly limited. This in turn translates into the risk of an overall decrease in mission effectiveness, due to insufficient or inconsistent training.

One approach to addressing this problem is to reduce the costs and increase the capabilities and effectiveness of modeling and simulation solutions for training. This paper addresses a portion of that approach, focused on providing "anytime/anywhere training". This can be achieved in part by providing knowledge-rich intelligent agents that replace human role-players and operators in training simulations, thereby reducing manpower costs and requirements. It can also be achieved by the portable design of intelligent agents, which abstract tactical reasoning processes from simulation-specific implementations. We have developed a variety of such agents for systems focused on training and experimentation with SOF skills in particular. This paper provides an overview of a subset of these agents, together with a description of our general approach and some of the technical advantages of using such agents.

Knowledge-rich agents are distinguished as being fully autonomous successors to Semi-Automated Forces (SAFs) to model human behavior in military simulation. Systems we have developed include TacAir-Soar, which controls fixed-wing aircraft flying a large variety of missions; SOF-Soar, a model of ground forces that perform reconnaissance, sniper, forward observer, patrol, and cordon/assault missions; Helo-Soar, an intelligent controller for rotary-wing aircraft flying assault, reconnaissance, and close-air support missions; and IF-Soar, which provides an indirect fire team to assist in training of forward observers for close-air support missions.

THE ARCHITECTURE

All of these systems use the same overall software architecture, which has evolved along with iterative refinements of the intelligent agents. An initial version of the architecture is described by Schwamb, Koss, & Keirsey (1994). For context, we provide here a brief overview of the current version of the architecture and its components. Figure 1 illustrates the interfaces between the simulation engine, the Soar cognitive architecture (Laird, Newell, & Rosenbloom, 1987), and the intelligent behaviors built into a particular agent (TacAir-Soar, in the example figure). The components are described individually below.

Simulation Engine

We have integrated our agent systems with a variety of simulation platforms, including JSAF, VR-Forces, and OneSAF, leading us to the development of a plug-in model for behavior model integration. We add a Simulation Abstraction Adapter to each simulator, which provides a uniform interface to the intelligent agents for sensing and control. The simulation engine provides terrain information, information about other entities, sensor and weapons models, as well as the physical models controlled by Soar agents. By providing a uniform interface to behavior models, we are able to move a model from one simulation

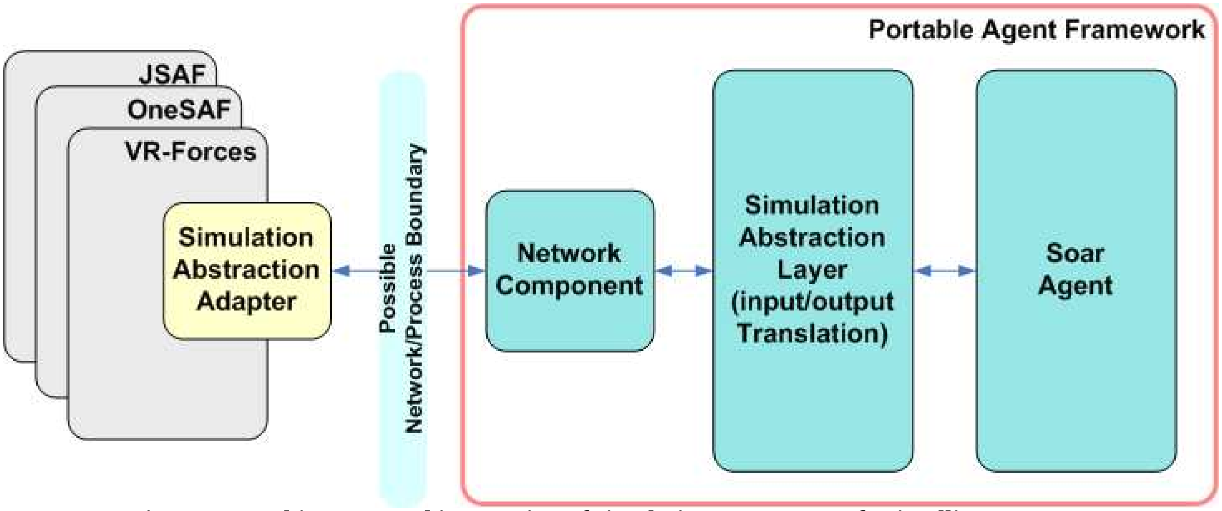


Figure 1. Architecture and integration of simulation components for intelligent agents.

environment to another with no changes to the model, aside from its particular interface to the simulator.

Soar

Soar is both a theory of human cognition and an embodiment of that theory in a programming architecture. Originally developed in 1982 at Carnegie Mellon University by Allan Newell and his students John Laird and Paul Rosenbloom, Soar has served worldwide as the basis of research in cognitive science, psychology and artificial intelligence, as well as the reasoning engine for some commercial applications. For more details Soar's history and architecture, see Laird, Newell, and Rosenbloom (1987).

Soar encodes an associative memory model with stimulus-response pattern-matching *production rules*. The rules are organized to frame decision-making as the selection and application of operators to achieve goals. The production rules represent long-term knowledge in the form of *if…then* statements whose *if* patterns match against a representation of the environment and the agent's own internal state. Actions serve to propose new operators, dynamically decompose more abstract operators, or send motor-control commands to the underlying simulation. Soar supports both goal-directed and reactive behavior, which makes it an ideal choice for implementing agents that must act in complex environments and realistic timeframes. The production rules implement an Abstract Decision Cycle, in which the agents repeatedly sense the environment, interpret new information to update the current representation of situational understanding, use the current understanding of the world to activate particular goals, and then select deliberate actions to achieve those goals (see Figure 2).

Agent/Simulation Abstraction

We have defined a plug-in architecture that abstracts the common interactions between simulations and intelligent agents, implementing them into an integrated set of reusable software components. The collection of components and interfaces constitute a Portable Agent Framework (PAF). PAF provides the interface components for the translation layer that exists between the simulation environment and the intelligent agent. PAF provides a plug-in environment that facilitates standardized communications between simulation components, including a simulation engine, agent systems (based on Soar or some other agent architecture), and associated tools, such as mission editors, graphical user interfaces, and speech interfaces. PAF enforces *Simulation Abstraction* that captures the most common types of interactions between agents and simulators, and is easily extensible to support new simulator or agent capabilities. The Simulation Abstraction Adapter converts information about the world into a form the agent can reason about, and converts agent actions into observable effects in the environment.

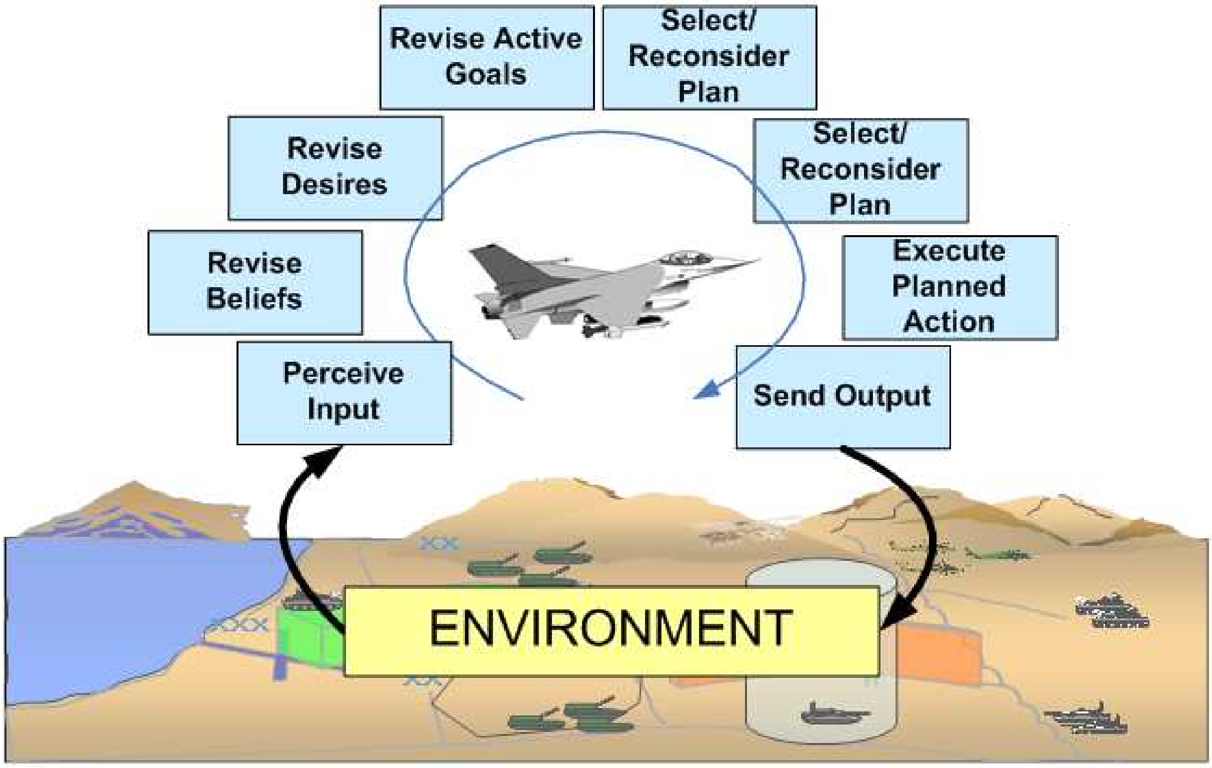


Figure 2. The Abstract Reasoning Cycle provides the context in which intelligent agents create situational understanding, activate task goals, and initiate actions to achieve those goals.

EXAMPLE AGENT APPLICATIONS

The following sections provide an overview of four different agent systems that are relevant to SOF training. This includes the systems mentioned above: TacAir-Soar, SOF-Soar, Helo-Soar and IF-Soar.

TacAir-Soar

TacAir-Soar is the first and largest agent system we have developed. It is an intelligent, rule-based system that generates believable "human-like" behavior for fixed-wing aircraft simulations. The application implements a number of innovations, including the scale of reasoning capabilities, integration with a rich and complex simulated environment, representation of human-like coordination and communication, and a rich implementation of situational understanding to drive agent reasoning. The system is capable of executing most of the airborne missions that the United States military flies in fixed-wing aircraft. It accomplishes this by integrating a wide variety of intelligent capabilities, including reasoning about interacting goals, reacting to rapid changes in real time (or faster), communicating and coordinating with other agents (including humans), maintaining situational awareness, and accepting new orders while in flight.

TacAir-Soar relies on mature intelligent systems technology, including a rule-based, hierarchical representation of goals and situation descriptions. Unlike many SAFs, the system does not just model a small set of tasks pertinent to military fixed-wing missions; it generates appropriate behavior for a broad variety of such missions routinely used by the US Navy, Air Force, and Marines; the UK Royal Air Force; and opponent forces in full-scale exercises. In addition to reasoning about complex sets of goals, the system coordinates and communicates with humans and other automated entities. The system must generate its behavior in real time (and sometimes faster). It must also integrate seamlessly into current military training exercises, and be able to cover unanticipated situations, so it does not interrupt the flow of training. Finally, all of the task requirements are set by existing military needs, and we were thus not able to tailor or simplify the domain to suit our purposes.

TacAir-Soar initially saw use in the Synthetic Theater Of War 1997 (STOW-97)/United Endeavor Advanced

Concept Technology Demonstration, an operational training exercise consisting of 48 straight hours and approximately 700 fixed-wing aircraft flights, all flown by instances of the TacAir-Soar system. In STOW-97, the emphasis was on providing tactical air-to-air and air-to-ground behaviors.

TacAir-Soar played important roles in exercises and demonstrations such as Roadrunner'98 and COYOTE'98 (Nielsen et al., 2000). TacAir-Soar was also indispensable in the Joint Forces Command's Joint Experiment '99 and Attack Operations '00, as well as many of the Navy's Fleet Battle Experiments. Additionally, TacAir-Soar was fielded as part of the Battle Force Tactical Trainer (BFTT) delivered to the Navy. Continued development of TacAir-Soar in recent years has adapted the system to support SOF training for close-air support missions and to support simulated experimentation with new UAV models. TacAir-Soar's inclusion in these projects has demonstrated one of the advantages Soar-based systems have shown over conventional SAFs—autonomy. A single operator can control hundreds of Soar agents, with intervention required only when the operator wants to change their mission details (Jones et al., 1999).

One of the most recent development emphases for TacAir-Soar has been on behaviors to support training Forward Observers (FOs) and Terminal Air Controllers (TACs) in Close Air Support (CAS) missions. For this application, the TacAir-Soar agents must include proper behaviors for flying all the CAS-supporting aircraft, in all phases of the mission, including communication and coordination knowledge for following proper procedures in interpreting the guidance of the TAC. This includes the ability to communicate using doctrinally correct speech, facilitated by off-the-shelf speech interface software. The trainee is responsible for identifying targets, communicating CAS mission information to the aircraft, and following through for re-attacks or release of the aircraft, all using a speech interface.

IF-Soar

IF-Soar is another system we have built with the goal of improving training for FOs and TACs performing indirect fire missions. Training Forward Observers in a live environment presents a wide range of logistical problems. Indirect fire missions involve a large physical area, as munitions are typically fired at targets beyond visual range. More importantly, there is a high cost to these exercises, as they require not only a great deal of materiel per mission but also other participants to play other roles, including the Fire Direction Center and the artillery batteries.

The use of a training simulation can alleviate many of these cost factors. Space constraints are reduced to the physical size of the simulation system, while virtual munitions that explode only on a computer screen eliminate the waste and danger associated with live rounds, and also the additional live persons required to fire them. However, such a system does not eliminate the requirement for other interactive participants. For example, a Fire Direction Center (FDC) must be present to handle the radio calls for fire from the FO. In the work on IF-Soar, we have developed an automated FDC that is capable of the following:

- Understanding and processing incoming Call For Fire (CFF) requests
- Producing doctrinally correct CFF acknowledgements and other required messages
- Reacting to errors or omissions in CFFs and interacting with the FO to identify them
- Implementing received CFFs in a simulated environment by tasking artillery batteries to fire

The primary emphasis of IF-Soar's design is on the appropriate spoken-word interactions between the IF team and the forward observer (FO). In training systems that employ IF-Soar agents, a human trainee (or operator) plays the role of the FO in charge of submitting CFFs. These calls are made over a voice interface and converted into a machine-readable form using a speech-recognition engine and grammar parser. The IF-Soar agent receiving these calls will then process the message, make any necessary modifications or additions to the fire mission, then generate a doctrinally correct spoken response to the FO. Once a CFF is completed, IF-Soar agents will carry out the firing order within the simulation environment.. The FO can then use the observed results of those fires to make adjustment fires based off previous CFFs.

Helo-Soar

Helo-Soar agents are designed to fly simulated rotary-wing aircraft. The initial knowledge base for Helo-Soar equipped the agents to fly Air Assault and Reconnaissance missions, as well as to participate in impromptu, on-call Medevac missions. From a user perspective, the primary emphasis on Helo-Soar's design was to provide an automated wingman capability, so that the agents would fly in groups that were led by a human flying a virtual simulator. Each synthetic wingman understands its role within the flight group and can reconfigure its position and role within the group as circumstances change. For example, a member of the group might run out of weapons or be destroyed, necessitating reconfiguration of the group's roles and command structure.

After receiving the mission brief, Helo-Soar agents plan and execute their missions using appropriate doctrine and tactics. They react to environmental changes, reforming to continue the mission on their own if the lead is disabled. The competence level of the agents increases the effectiveness of trade studies and training simulations. This version of Helo-Soar may be configured for a single wingman or multi-aircraft formation flight behind a human lead in a synthetic cockpit. The lead re-directs the aircraft in flight using voice commands, and receives replies and status updates over a radio headset. The agents are briefed with the same information as the human pilot, and have demonstrated the ability to continue the mission in the event that the lead aircraft is lost.

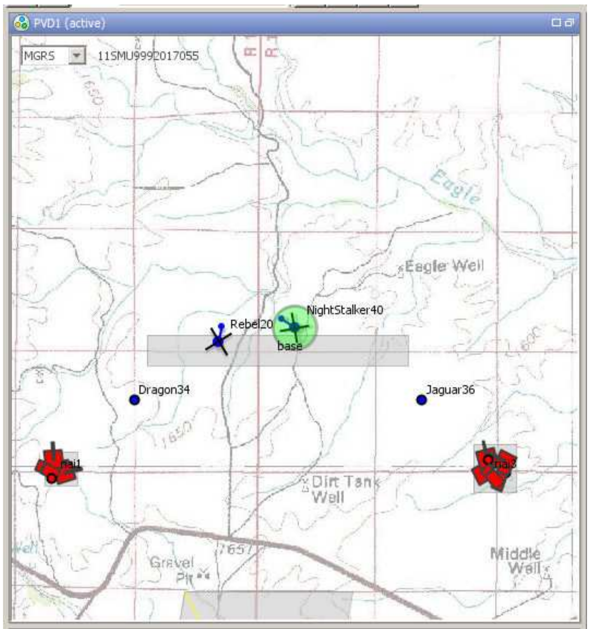


Figure 3. Two Helo-Soar agents flying CAS missions. The one on the right is being controlled by a SOF-Soar agent performing as a TAC.

Extensions to Helo-Soar have added the ability to perform close-air support missions. These capabilities were added in the context of the close-air support training system that also included TacAir-Soar and IF-Soar agents. Figure 3 shows a snapshot of two groups of Helo-Soar agents performing a CAS mission. The group on the left is under the control of a human TAC, guiding the Helo-Soar missions with a simulated radio that uses a speech-recognition system to translate commands. The group on the left is under the control of another intelligent agent that knows how to spot targets and call in CAS strikes. This TAC agent is described further in the following section.

SOF-Soar

The final example agent system we describe is SOF-Soar, which has a slightly different emphasis from the previous two systems. Where TacAir-Soar and IF-Soar are intended to fill non-SOF roles in the training of SOF personnel, SOF-Soar agents actually take the place of SOF elements within the simulation. This can be to populate a training environment with a rich set of elements to complete an exercise scenario, or it could be to provide teammates with which human SOF trainees can interact. SOF-Soar operates in a similar manner to previously described Soar systems in that each agent represents a single SOF operator who can observe the environment, pursue mission-relevant goals, and respond to threats. In the case of SOF-Soar, there is an added focus on multi-agent teams where each agent can have a specific role within the team. Furthermore, differences lie in the kinds of missions and behaviors implemented, such as reconnaissance (shown in Figure 4), urban patrol, and forward observer missions.

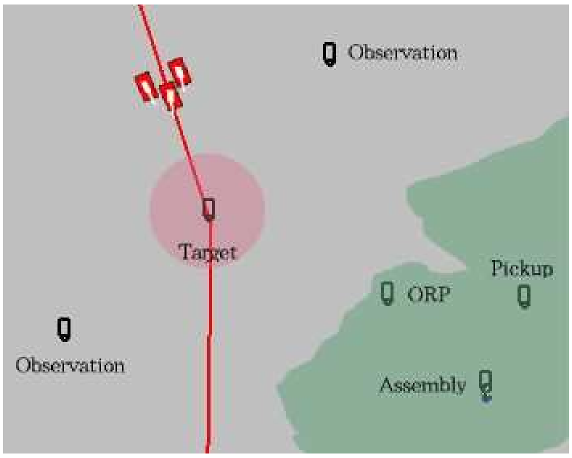
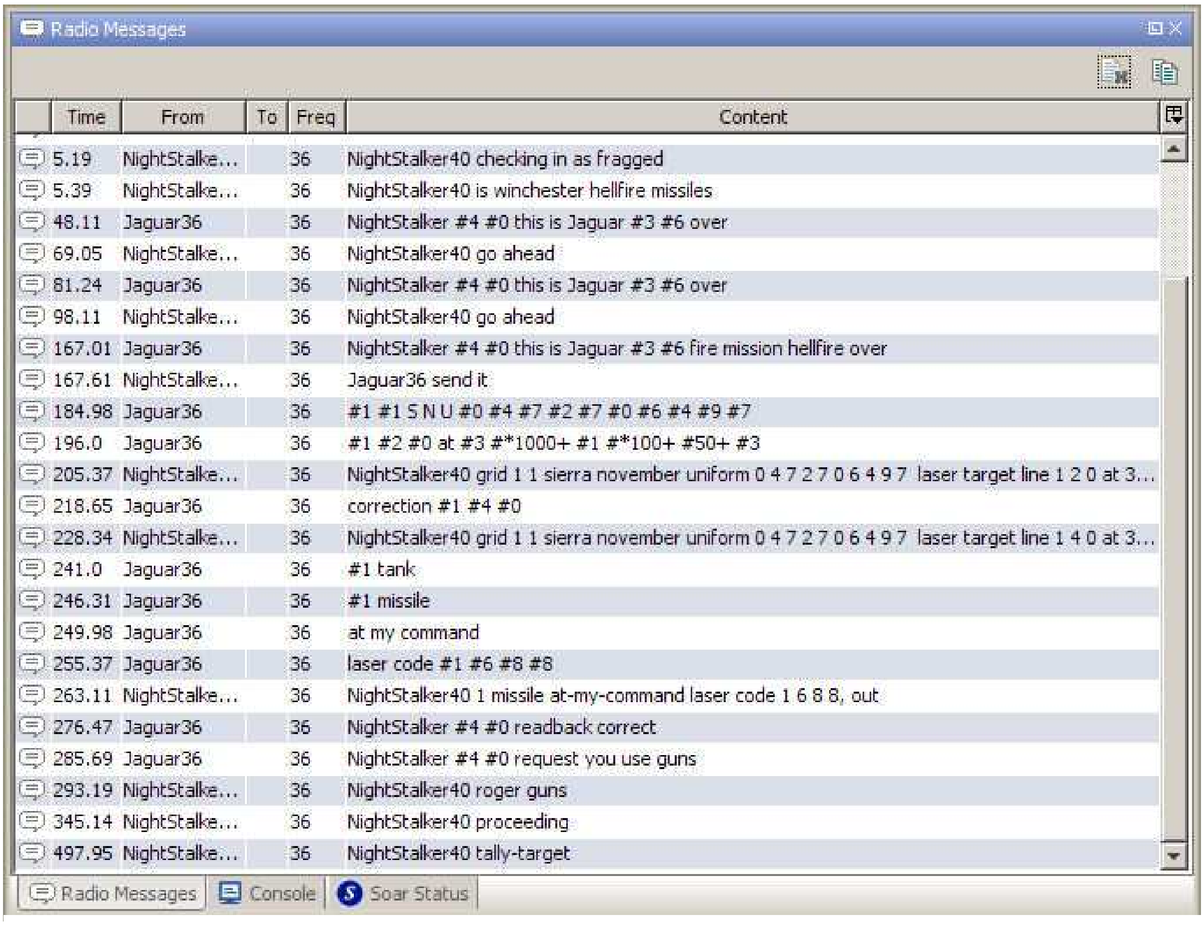


Figure 4. SOF-Soar agents performing a long-range reconnaissance mission.

In forward observer missions, SOF-Soar agents provide the role of forward observers in a simulated environment. These agents provide the complementary role to the TacAir-Soar and Helo-Soar agents described earlier, where the SOF-Soar entities identify relevant targets, communicate target information using doctrinally correct radio calls to the aircraft, and follow through with requests for re-attack or release of the



Radio Messages

Time	From	To	Freq	Content
5.19	NightStalke...		36	NightStalker40 checking in as fragged
5.39	NightStalke...		36	NightStalker40 is winchester hellfire missiles
48.11	Jaguar36		36	NightStalker #4 #0 this is Jaguar #3 #6 over
69.05	NightStalke...		36	NightStalker40 go ahead
81.24	Jaguar36		36	NightStalker #4 #0 this is Jaguar #3 #6 over
98.11	NightStalke...		36	NightStalker40 go ahead
167.01	Jaguar36		36	NightStalker #4 #0 this is Jaguar #3 #6 fire mission hellfire over
167.61	NightStalke...		36	Jaguar36 send it
184.98	Jaguar36		36	#1 #1 S N U #0 #4 #7 #2 #7 #0 #6 #4 #9 #7
196.0	Jaguar36		36	#1 #2 #0 at #3 #*1000+ #1 #*100+ #50+ #3
205.37	NightStalke...		36	NightStalker40 grid 1 1 sierra november uniform 0 4 7 2 7 0 6 4 9 7 laser target line 1 2 0 at 3...
218.65	Jaguar36		36	correction #1 #4 #0
228.34	NightStalke...		36	NightStalker40 grid 1 1 sierra november uniform 0 4 7 2 7 0 6 4 9 7 laser target line 1 4 0 at 3...
241.0	Jaguar36		36	#1 tank
246.31	Jaguar36		36	#1 missile
249.98	Jaguar36		36	at my command
255.37	Jaguar36		36	laser code #1 #6 #8 #8
263.11	NightStalke...		36	NightStalker40 1 missile at-my-command laser code 1 6 8 8, out
276.47	Jaguar36		36	NightStalker #4 #0 readback correct
285.69	Jaguar36		36	NightStalker #4 #0 request you use guns
293.19	NightStalke...		36	NightStalker40 roger guns
345.14	NightStalke...		36	NightStalker40 proceeding
497.95	NightStalke...		36	NightStalker40 tally-target

Radio Messages | Console | Soar Status

Figure 5. Sample communication interaction for a CAS mission flown by Helo-Soar, with TAC function provided by SOF-Soar.

aircraft. Combined with TacAir-Soar and Helo-Soar agents, these SOF-Soar observers allow for a fully-automated CAS capability into a simulation environment, for analysis or experimentation. However, because the SOF-Soar agents are using doctrinally correct communications, they could similarly interact with human pilots, facilitating the training of pilots for CAS missions. Figure 5 provides a sample dialog for a CAS mission flown by Helo-Soar under the control of SOF-Soar.

AGENT-HUMAN COMMUNICATION

A key advantage to intelligent agents over competing simulation technologies is the ability to communicate and coordinate naturally with humans participating in the simulation (whether the humans are operators, role players, trainees, etc.). Communication in particular is important because it enables agents to coordinate their actions verbally with humans and other intelligent agents (Nielsen et al., 2000). For most applications, our Soar-based agents use simulated radios as their primary form of communication. Each radio is assigned a particular frequency, and all agents that have a radio tuned to that frequency, within the range of that transmitter, can “hear” the broadcast. The agents can direct their messages to individual recipients by prefacing each message with the name of the intended recipient. It is up to the receiver to process or disregard the message based on the named recipient.

As with human-human interactions, human-agent interactions can take place via different modes of communication. Sometimes there are formal interfaces and channels of communicate (such as fixed-format mission briefs), and in other cases humans talk to each other using natural language, but constrained by doctrinal military grammars. In addition to the agent systems described above, we have built associated tools that support various modes of interaction. Here we describe three of these tools: the communications panel, exercise editors, and speech recognition.

Communications Panel

For the purposes of running exercises, we allow human simulation operators to communicate directly with individual agents. A tool called the Communications Panel (or Comm Panel) enables an operator to tell the agents to change their mission parameters during mission execution. The agents receive the commands as human-readable text messages on their radios. The Comm Panel provides templates to construct all of the message types that the intelligent agents understand, and it serves as an inexpensive and reliable replacement to speech-recognition systems, when such systems are not available or not suitable.

Exercise Editors

For rapid exercise development, we have developed mission specification tools for each type agent. We call these tools exercise editors, because they specify all of the exercise-relevant information that an agent needs to know to complete its mission. For humans, this would be the information they receive in their mission briefs, combined with information about standard operating procedures, rules of engagement, etc. The exercise editors allow users to specify all the pre-briefed information the agents expect or require, facilitating the generation of large exercises in short periods of time. As the number of missions performed by the agents grows, we similarly expand the associated exercise editors to accommodate the new mission types. Because the agents are fully autonomous, the information provided by the exercise editor is all they need to know to perform their missions.

Speech recognition and dialog management

The most sophisticated form of human-agent interaction relies of spoken voice communication. For the agents we have described hear, we support human speech recognition using the SoarSpeak appliance, first introduced by Jones, Nielsen, and Taylor (2000). SoarSpeak is a collection of systems that provide text-to-speech (TTS) and speech-to-text (STT) support for Soar agents that operate within HLA or DIS-compliant simulation environments. SoarSpeak generically wraps commercial off-the-shelf STT and TTS systems to provide a consistent interface for including speech in an application. In the case of grammar-based recognition, SoarSpeak also is capable of generating a semantic parse of an accepted utterance, rather than just the raw utterance string.

The Soar-based agents use grammar-based recognition to recognize, understand, and respond to utterances spoken by human participants in the simulation, such as the forward observer, AWACS controllers, or simulation operators. The result of an accepted utterance (one that is grammatically correct) is a *semantic parse* in a text-based XML format that assigns semantic value to the contents of the utterance. For example, the message *"alpha 3 romeo 5 1 this is hopper, over"* would result in the following semantic parse that would be passed into an agent:

<you-this-is-me>
  <msg-type> introduction </msg-type>
  <you> ***A3R51*** *</you>*
  <me> ***hopper*** *</me>*
</you-this-is-me>

SITUATIONAL AWARENESS

A key distinction between intelligent agents and SAF agents is the matter of situational awareness and understanding. While all of these systems are engineered to produce particular responses in particular situations, there is a significant difference in the underlying representations and approaches used to generate the responses. The intelligent agents we have described to not try to do anything until they have first interpreted and understood the situation they are sensing. This includes maintaining a record of past significant events and goals. The Soar-based agents obtain information about the environment from sensors, communications, and the pre-briefed mission specification. When they become aware of an entity in the simulation, they can deliberately seek more information by focusing their attention on that specific target. If the entity has gone out of the range of the model's sensors, information about that entity remains in attentional memory for a set amount of time before being dropped; the agent will completely forget about the entity unless the agent has deliberately decided to remember it. (This same forgetting mechanism holds for communication as well.) Similarly, if an agent has lost contact with an entity it is aware of, the agent projects the location of the agent while it is not directly sensed. If the agent has lost contact with the entity for an extended period of time, the agent deliberately forgets about it.

We have developed a tool called the Situational Awareness Panel to provide graphical visualizations of the understanding of the intelligent agents (Jones, 1999; Taylor et al., 2002). This tools presents information about what the agent senses, as well as its internal state information, the current goals the agent is pursuing, and important milestones achieved during the course of the mission. The graphical display of this information allows human inspection of the agent's reasoning process, so users can understand why the agents are making the decision they are making, and what the agents believe about the state of the environment.

PORTABLE BEHAVIORS AND DATA

A final key aspect of our approach to intelligent agent design is the focus on the portability of behaviors and knowledge representation. Most SAF systems are intimately integrated with a particular simulation engine and technology. This means, for example, that there is no cost-effective way to port a behavior model from JSAF into OneSAF, because the implementation of the model is so closely tied to the implementation of the simulator. Our approach has been to abstract the "minds" of the intelligent agents from the physical platforms and environment with which the agents interact. Although JSAF and OneSAF (just to take two examples) each have their own implementations of the world and the task environment, they are conceptually trying to represent the same military domains, missions, and tasks. Thus, at least to a large extent, an agent model for one simulator should be performing the same types of reasoning as it would for the other simulator. Our approach is to implement the "mind" of each intelligent agent just once, and then account for any simulator-specific assumption in the Simulation Abstraction Adapters required by the ATE plug-in environment.

Using this approach, we have demonstrated a number of advantages of intelligent agents over standard CGF technology. Primary among these are:

- A level of intelligent capability that can reduce the required number of human operators by two or more orders of magnitude.
- A simulation-neutral design and implementation that allows Soar Technology CGFs to integrate with a variety of simulation platforms. Integrations that we have produced so far include JSAF, OneSAF, OTB, STAGE, and VR-Forces, among others.

Key to the ability to port the intelligent agents to a variety of platforms is our development of a platform-neutral language for describing mission parameters from the exercise level (e.g., rules of engagement, weather, and waypoints) through the mission level (e.g., commit criteria, routes, and intercept tactics) and ultimately to the element level (e.g., call signs, force structure, weapon loadouts, and radio frequencies). We have encapsulated these abstractions into the exercise editors for each agent type (e.g., Coulter & Laird, 1996). Current work focuses on formalizing these abstractions into an XML-based, platform-neutral representation of knowledge, missions, and behaviors. The combination of a portable representation for CGF behaviors and a portable data format for mission parameters should provide a future cost-effective path for adoption of intelligent agents across a variety of simulation environments.

CONCLUSION

We have described an intelligent agent technology that has the potential to increase the cost effectiveness of SOF training in modeling and simulation, as well as to provide new training capabilities, such as structured doctrinal grammars using speech-recognition systems. The increases in cost effectiveness, along a variety of dimensions, should in turn increase the number of mission-critical experiences that SOF warfighters can be exposed to, ultimately leading to a more effective fighting force.

We have additionally described four examples of intelligent agent applications we have developed that are relevant to SOF training. We intend these examples to highlight the types of capabilities that intelligent agents can provide to the training environment. We have emphasized three keys to the strengths of intelligent agents for SOF training:

- Intelligent agents base their decision making on situational awareness and understanding. Every choice is made in the context of a doctrinally consistent interpretation of the current situation.
- Intelligent agents have the capability to coordinate and communicate with humans (and each other) using doctrinally correct grammars and modes of interaction. This provides the ability to avoid negative training by allowing the SOF trainees to use the modes of interaction they will actually use in the operational environment.
- Intelligent agents are defined in a simulation-independent fashion, focusing on the reasoning required to complete the mission rather than the specific aspects of the underlying simulation implementation. In the long run, this will allow more cost effective deployment of agent systems and migration to new simulation platforms as they are developed.

REFERENCES

Calder, R., Smith, J., Courtemanche, A., Mar, J., & Ceranowicz, A. (1994). ModSAF Behavioral Simulation and Control. Proceedings of 3rd CGF&BR.

Coulter, K. J., & Laird, J. E. (1996). A Briefing-based Graphical Interface for Exercise Specification. Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation. pp. 113-117, Orlando, FL.

Jones, R. (1999). Graphical Visualization of Situational Awareness and Mental State for Intelligent Computer Generated Forces. Proceedings of the 8th CGF&BR.

Jones, R.M, Laird, J.E., & Nielsen, P.E., Coulter, K.J., Kenny, P.K., & Koss, F.V. (1999). Automated Intelligent Pilots for Combat Flight Simulation. AI Magazine 20(1), 27-42.

Jones, R., Koss, F., Nielsen, P. & Taylor, G. (2000). Communication with Intelligent Agents. Proceedings of the 22nd Interservice/Industry Training Systems and Education Conference (I/ITSEC), Orlando, FL.

Laird, J., Newell, A., & Rosembloom, P. (1987). Soar: An Architecture for General Intelligence. Artificial Intelligence 33(1), 1-64.

Nielsen, P.E., Koss, F., Taylor, G., & Jones, R.M. (2000). Communication with Intelligent Agents. I/ITSEC.

Nielsen, P.E., Smoot, D., Martinez, R., & Dennison, J.D. (2000). Participation of TacAir-Soar in RoadRunner and Coyote Exercises at AFRL, Mesa, AZ. *Proceedings of 9th CGF&BR*, pp. 173-180.

Reece, D., Qurston, D., Kraus, M., & Carbia, I. (1998). Updating ModSAF for Individual Combatants: the DI SAF Program. Proceedings of the 7th CGF&BR.

Schwamb, K., Koss, F., & Keirsey, D. (1994). Working with ModSAF: Interfaces for Programs and Users. Proceedings of the 4th CGF&BR.

Taylor, G., Jones, R. M., Goldstein, M., Frederiksen, R., & Wray, R. E. (2002). VISTA: A Generic Toolkit for Visualizing Agent Behavior. *Eleventh Conference on Computer Generated Forces and Behavior Representation*, Orlando, FL, University of Central Florida