Communication with Intelligent Agents

Paul Nielsen, Frank Koss, Glenn Taylor, Randolph M. Jones
Soar Technology, Inc.
317 N. First St., Ann Arbor, MI 48103-3301
{nielsen, koss, glenn, rjones}@soartech.com

Abstract

Combat effectiveness requires coordination, and the most critical component of coordination is communication. This paper describes methods for communication with and among intelligent constructive forces (IFORs). It explores a number of different approaches to communication and their implementation in the TacAir-Soar behavior system.

Because TacAir-Soar entities are intended to be indistinguishable from human combatants within the simulation environment, communication may occur between IFORs, between humans and IFORs, and between IFORs and other constructive forces. This places a number of constraints on the possible forms of communication. They must be natural for human interaction, yet well structured for communication with other constructive forces.

TacAir-Soar s approach to communication is to model mechanisms used in the real world rather than to create simulation-specific versions. For example, radio messages are text representations of the same, doctrinally correct, English utterances spoken by human pilots and controllers. The resultant system makes it comparably easy to interchange roles between humans and IFORs. This approach also facilitates interaction with other constructive forces because of well-defined communication templates and optional translation to the Command and Control Simulation Interface Language (CCSIL) [Salisbury, 1995].

TacAir-Soar includes a range of communication methods, including explicit and implicit forms of communication. The methods presented here include natural language communication over simulated radios; a communication panel and radio log for graphically driven communication; SoarSpeak, for real-time speech recognition and generation; distributed goal and status reports for communication with controllers; data links; non-verbal communication; and translation methods for CCSIL.

This paper examines how these various modes are implemented, and their benefits and drawbacks. Specifically, it shows how the implementations enable humans to easily immerse themselves into a simulation involving IFORs. It also includes several examples drawn from technology demonstrations and operational exercises where humans communicated with IFORs serving as command and control entities, friendly forces, and wingmen.

Biographical Sketches:

Paul E. Nielsen Senior Scientist and co-founder Soar Technology, Inc. He previously was an Assistant Research Scientist at the Artificial Intelligence Laboratory of the University of Michigan where he was integrally involved in the DARPA-funded research that resulted in TacAir-Soar. Prior to joining the University of Michigan he worked at the GE Corporate Research and Development. Dr. Nielsen received a Ph.D. in Computer Science from the University of Illinois in 1988.

Glenn Taylor, Systems Engineer, has over five years of industry experience in Software Engineering. He works in developing behaviors for intelligent agents, as well as tools to support intelligent agent technology. Prior to joining Soar Technology, Inc., Mr. Taylor worked in Ann Arbor, MI for C-TAD Systems, Inc., writing CAD-conferencing software, and for JHA Simulations, Inc., writing training simulators for the heavy industry. He has a B.S. and in Computer Science, and an M.S. in Computer Science and Engineering, both from the University of Michigan.

Frank V. Koss, Senior Systems Engineer, has over five years' experience in designing and implementing the underlying interface between Soar and simulation systems. His experience covers implementing major changes to the interface, including integrating a new version of Soar and responding to significant changes to fundamental structures of the simulation system. He has an M.S.E. in Computer Science from the University of Michigan, and a B.S. in Computer Engineering at Carnegie Mellon. He is Project Manager for Soar Technology's contracts under MSIAC for modeling special operations forces for JFCOM/DARPA, and CGF support for JFCOM's AO00 Exercises and USN NWDC's Fleet Battle Experiments.

Randolph M. Jones is a Senior Scientist and Vice President at Soar Technology, Inc. He is also an Assistant Professor of Computer Science at Colby College. In 1989, he received his Ph.D. in Information and Computer Science from the University of California, Irvine. His general areas of research and development include computational models of human learning and problem solving, executable psychological models, and automated intelligent actors for virtual environments. He has worked on the TacAir-Soar project since its inception, and wrote the first implementation of the TacAir-Soar system.

Communication with Intelligent Agents

Paul Nielsen, Frank Koss, Glenn Taylor, Randolph M. Jones
Soar Technology, Inc.
317 N. First St., Ann Arbor, MI 48103-3301
{nielsen, koss, glenn, rjones}@soartech.com

INTRODUCTION

To model human behaviors accurately in synthetic environments, computer generated forces (CGFs) need to coordinate their actions through communication in order to achieve both individual and collective goals. To do this they must be able to translate internal information about goals and states into a form that can be understood by others, and they must translate messages from other entities into an appropriate internal representation. Further, if CGFs are to interact with humans, who already have well-developed languages for communication, CGF communication must be similar to human communication. It must be easy to understand and generate, it must provide neither more nor less information than actual human communication, and it must facilitate simulation applications that emphasize communication, such as information warfare.

For those reasons, CGFs need to emulate real-world communication procedures to the level of accuracy required for effective military coordination. They need to interact with command and control through proper communication operations as well as other information networks. If we are to follow the maxim Train the way you fight, training an operator in use of communication for simulation should be no different than training for real world communication.

Background

TacAir-Soar emulates human behaviors for pilots and controllers in the military, fixed-wing aviation domain. It has been under development since 1992, and has matured to the point where it currently provides the most robust synthetic force model for a number of military applications and behavior modeling research efforts [Jones et al., 1999; Jones et al., 2000; Nielsen et. al., 2000].

The goal of TacAir-Soar intelligent constructive forces is to develop human-like synthetic entities for populating simulation environments. In contrast to semi-automated forces, where it is assumed some higher level entity will be responsible for decisions requiring judgement, the TacAir-Soar approach is to endow all entities with knowledge, situational awareness, and decision making abilities similar to humans performing similar tasks. The validity of this approach has been confirmed in part by participation in numerous military events. These intelligent forces provide a payoff in terms of increasing the fidelity of the entity's behavior, while decreasing the demand for operator supervision. [Laird et al., 1995; Jones et al., 1999]

Each TacAir-Soar agent represents a person, pilot or controller, performing its mission role as part of a scenario. These scenarios may range from one-on-one engagements to multi-ship coordinated missions to theater level operations involving hundreds or thousands of aircraft.

What distinguishes TacAir-Soar from other approaches to CGFs is that the entities are truly autonomous and unscripted. Once tasked, they carry out their assigned missions, react to threats, coordinate their actions, and communicate with other entities (other flight members, or Command and Control operators) without requiring any human intervention. If human intervention is desired, it happens in the way real-world interactions occur, through mission task orders transmitted on (simulated) radios or data links.

TacAir-Soar accomplishes this by integrating a wide range of intelligent capabilities, including real-time hierarchical execution of complex goals and plans, communication and coordination with humans and simulated entities, maintenance of situational awareness, and ability to accept new orders while in flight. [Jones et. al., 1999]

TacAir-Soar was originally designed for large-scale operations (i.e., theater-level with thousands of units), in a completely synthetic environment. In STOW 97, TacAir-Soar demonstrated the ability to generate autonomous, real-time, high fidelity behavior for a large-scale simulation of a complete theater battle. [Jones et al., 1999] Since then it has been used in such diverse applications as pilot training at AFRL, Mesa [Nielsen et. al., 2000]; air controller training as part of the USN Battle Force Tactical Trainer Air Management Node [Jones & Kenny, 2000]; joint training as part of the JFCOM J-9 Warfighting Experiment; and air support for the USN Fleet Battle Experiments.

APPROACH

To the greatest extent possible, TacAir-Soar attempts to model communications as they occur in the real world. For example, the system uses English sentences from Comm Brevity terminology, doctrine, and standardized reports. In part because current bandwidth limitations make it infeasible to transmit audio signals between entities, the current system interface uses speech-to-text/text-to-speech conversion and transmits text strings.

TacAir-Soar models a wide range of communication modes including voice, language, visual cues, and data communication. One important tenet in the military domain is to minimize communication. Each transmission affords a chance of detection, adds to battlefield confusion, and clutters bandwidth. Yet, as we assert above, communication is essential for effective coordination. One way to avoid this dilemma is through comprehensive preplanned information.

A second way to minimize communication is through the use of visual cues. For example, the aircraft do not have to announce their turns - they just turn and the wingmen follow. Yet, in a dynamic flexible environment, like a battlefield situation, the need for effective, explicit communication cannot be ignored. Thus, humans and TacAir-Soar supplement explicit communication methods with common knowledge for coordination and standard operating procedures [Laird, Jones, & Nielsen, 1998].

LANGUAGE

For many applications, the ideal mode of communication between humans and synthetic forces is natural language. However, the problems of natural speech recognition and generation have not yet been solved. It is ambiguous, difficult to parse, and difficult to understand. For these reasons, most CGF implementations avoid it altogether. Even for text-based messages, full natural language understanding is still beyond the state of the art. However, we are fortunate in this domain to enjoy well-structured communication procedures dictated by military doctrine. The standardized communication doctrine reduces ambiguity, because military personnel are taught what to say, how to say it, and when to speak.

TacAir-Soar agents currently use a large set of language templates that combine to allow relatively flexible, semi-natural recognition and generation of strings of words. These templates provide the basic language for communication between synthetic agents, as well as between agents and humans. This template-based approach prespecifies the form of the messages that the system can generate and accept. The agents know when to generate these messages, and how to interpret these messages and modify their own internal knowledge structures appropriately. TacAir-Soar contains over two hundred such templates. For example, the message to hand off an aircraft to another controller might be

check in with *<controller-name>* on *<radio-frequency>* at *<location>*

where the words in brackets represent variables.

Message generation consists of binding variable values and transmitting the result in the form of a string. Message understanding consists of accepting such a string and matching it against the keywords of the possible templates. If the keywords match, the values in the string are bound to the variables in the template. This method works for most cases; in very few cases additional testing is needed for value types or context.

Messages can range from simple, short communications with no variables such as

bogey dope

to more complex constructs with multiple variable bindings such as

<callsign>* group *<group-id>* *<number-of>* contacts bearing *<bearing>* range *<range>* miles bullseye heading *<heading>

The advantages of template based pattern matching are that it is relatively flexible, fast, simple, and understandable by humans. It separates the task of language understanding from the task of speech understanding and generation. But it is not real English: it cannot understand arbitrary phrases; it cannot understand paraphrases, unless they are explicitly represented in the templates; nor can it understand partial messages.

CCSIL

Template based parsing works for communication between TacAir-Soar agents because they all speak the same language. However, the problem of communication is further confounded by the need to send and receive messages from other CGFs that do not possess even this basic level of natural language capability. To address this problem, the agents also have knowledge of CCSIL [Salisbury, 1995]. CCSIL is essentially an enumerated set of data structures for military communications. For example, the directive to hand off an aircraft to another controller would be message 1107 with three data fields specifying controller, frequency, and location.

TacAir-Soar provides translation to and from these structures and the human understandable text strings described above. The translation process is an extra layer of pattern matching that matches the text strings on one side of a rule and produces CCSIL on the other side, or vice versa. As far as a human can tell, the agents are communicating in English, but between non-TacAir-Soar CGFs they are using highly structured CCSIL.

SPEECH GENERATION AND SYNTHESIS

In an effort to make communication between humans and TacAir-Soar agents more natural, we have developed SoarSpeak, a voice recognition and generation system based on IBM s commercial ViaVoice package [IBM, 2000]. SoarSpeak allows controllers and other pilots to speak directives to TacAir-Soar agents, eliminating the need for cumbersome keyboard or user-interface input. In some cases, it also eliminates the need for a person sitting between a trainee and the simulation, for example to translate directives from a human pilot to a synthetic wingman.

SoarSpeak Architecture

We faced two technical challenges in developing SoarSpeak. The first was interfacing ViaVoice to TacAir-Soar, and the second was developing an effective grammar for recognition.

ViaVoice provides an engine for recognition and generation, as well as a programming interface to that engine. At the time of the initial development of SoarSpeak, ViaVoice only ran under Windows operating systems. However, TacAir-Soar agents exist as part of the JSAF simulation system, which currently runs under the Linux operating system. These platform constraints were a driving force in the development of the SoarSpeak architecture, pushing us toward a socket-based communication paradigm.

We approached recognition and generation as different problems, and created two distinct applications to deal with these. Both use ViaVoice as their engine, but running as separate processes. The SoarSpeak recognition application consists of a simple GUI for selecting which grammar to use, a socket framework for communicating with the simulation, and the ViaVoice engine. The application makes calls to the ViaVoice API to obtain the results of parsing. When the engine recognizes a spoken phrase, it sends a corresponding annotated text string to the simulation. For instance, in directing an aircraft a user might utter,

hornet1 this is alpha-bravo vector 0 9 0

Before being sent over the socket, SoarSpeak marks the text with the radio and the frequency, as well as special end markers (MSG and #):

“MSG radio-a 2000000 hornet1 this-is alpha-bravo vector 090 #”

SoarSpeak sends the resulting string through a network socket to a waiting TacAir-Soar agent.

In the simulation, the SoarSpeak library creates a socket for listening, which is then polled by any agent designated as a SoarSpeak agent. The first to see the message plucks it off the socket and transmits the message over the agent s simulated radios (translating it into CCSIL when appropriate).

A SoarSpeak agent is defined simply as any TacAir-Soar agent that has loaded a specialized behavior set for interacting with the SoarSpeak library. If we desired higher bandwidth for messages, multiple agents could be endowed with these behaviors, so the responsibility for processing and transmitting messages could be spread over multiple agents, reducing the overhead for a single agent. It is also because of this overhead that not every agent is given this ability. For instance, if a TacAir-Soar agent were in the middle of cognitively intensive activities such as a dogfight, we would not want that agent to have to take the time to process incoming messages that it otherwise would not care about.

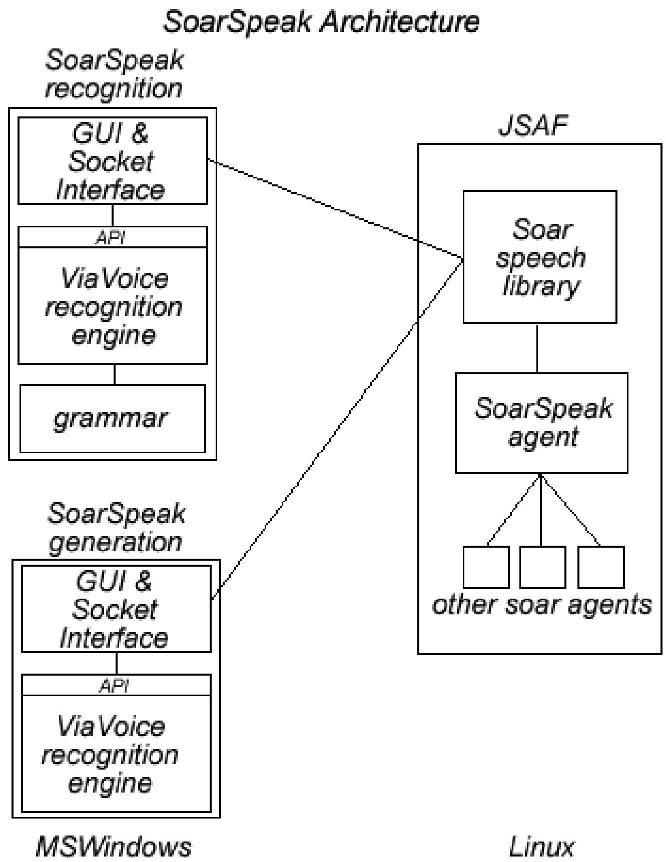

Figure 1 The SoarSpeak Architecture

In order to generate speech from text messages, the user chooses on the client side the agents to listen to (called tapping the agent), and can vary features of the voice (such as pitch and tempo) to make each agent sound distinct. The client also opens a socket and waits for messages from TacAir-Soar. On the simulation side, the SoarSpeak agent passes all radio communication to the Soar Speech library, which looks only for those messages sent by tapped agents. The raw text from a tapped agent is then sent across through a network socket connection to the SoarSpeak application for conversion to speech.

The second challenge was developing the grammar and vocabulary for the recognition system. We understand the difficulty in developing a full-fledged speech understanding system. The military air domain affords us some relief from the total problem: military communication is typically very regimented, allowing us to use simpler methods for recognition. We have solved this more limited problem using relatively simple grammars to obtain from the recognition engine complete phrases, which are processed by the TacAir-Soar agents using the same template-based matching scheme used to process any normal radio communication. To further reduce complexity, we developed different grammars for different domains; the only values that need to be changed within a domain tend to be the call-signs and the names of the waypoints in the scenarios. These can be entered into the grammar by the user, converted to ViaVoice s format off-line, and loaded into SoarSpeak while the exercise runs. These simplifications greatly aid the ViaVoice software in resolving ambiguous or unclear speech patterns.

Benefits

The benefits of speech recognition and generation technology are obvious in the broad sense: they allow hands-off use of a system, avoiding the time and expense of designing effective alternative user interfaces, and they make interaction between humans and computers more natural. SoarSpeak makes great strides in both of these categories. Its first use was at COYOTE 98 [Nielsen, et al., 2000], to facilitate using TacAir-Soar agents as synthetic wingmen to human pilots in training.

Additionally, SoarSpeak demonstrates the efficacy of building usable tools using COTS packages. It was designed for direct comparison to a much more expensive, custom-built speech recognition system, to prove that inexpensive solutions can also be effective solutions.

Issues

ViaVoice recognition is not 100% accurate. Users will, in the course of normal conversation, insert utterances such as um without realizing it, or pause to reorient themselves in the middle of a command. ViaVoice will sometimes interpret long mid-command pauses as the end of input, and try to process the phrase it has, typically rejecting it as unacceptable, thus forcing the user to start at the beginning of the utterance. This is especially true in very long utterances such as

"Tomcat recovery fixed wing ship heading 0 0 0 marshal on the 1 8 0 for nine take angels ten approach time thirty five."

The longer the utterance, the more likely the user will pause or insert extra sounds, decreasing the likelihood of recognition.

As any other communication in TacAir-Soar, messages are processed using template matching, which constrains the range of understood messages. Additionally, ViaVoice processes grammars described in Backaus-Naur Form (BNF), which is then converted to an internal format using an off-line translation program. This allows new grammars to be loaded on the fly, without the need to recompile SoarSpeak. BNF, however, also limits the expressability of the grammar. Moving away a template-based matching approach toward using the ViaVoice engine as simply a word recognizer, and pushing the recognition to a more robust parsing system, can overcome some of these problems. Alternatively, there may be ways to engineer the grammars to recognize more flexible sentence fragments. Further, developing more flexible natural language understanding capabilities in the agents to include much richer semantic and discourse processing would improve the believability of the agents. However, we understand the difficulty of the problem, and, while the domain of doctrinally correct pilot communication is a restricted one, general natural language understanding issues still need to be addressed.

Despite advances in voice recognition technology, commercial voice generation technology is still not terribly mature: the voices typically sound like a computer. Users have indicated that this is not entirely bad in some training situations, it is useful to know which entities in the simulation are humans and which are synthetic. However, in light of our overall goal of indistinguishability between TacAir-Soar agents and other entities in simulation, this remains an issue. An initial solution to this problem was to pre-record different human voices speaking a small range of responses and commands, but the number of required responses quickly outgrew the effectiveness of this solution. Still, it may be worth further exploring the use of digital voice sampling as opposed to voice synthesis.

Another hindrance to the usefulness of SoarSpeak is that it requires two machines to run: one for the SoarSpeak applications, one for the simulation. This will change once SoarSpeak is ported to the Linux operating system, made possible by the recent porting of ViaVoice to Linux.

GRAPHICALLY DRIVEN COMMUNICATION

Speech is just one mode of expression for language. As we have already discussed, the SoarSpeak system uses COTS speech recognition and generation software to translate between utterances and TacAir-Soar's template-driven internal language. But because the language module stands on its own, we have also been able to develop independent, alternative implementations of language expression.

Communication panel

One versatile tool that has seen relatively widespread use is the Communications Panel (CP) [Jones, 1998]. The CP is a menu-driven graphical interface that allows humans to compose language templates to send to synthetic agents. The CP contains almost all of the templates that TacAir-Soar agents understand, and categorizes the templates into groups, to provide easy and efficient access by a human user. The human user composes a message first by selecting a message category. Then, the human uses a menu button to select a particular message template (the menu labels each template with a meaningful name, such as "intercept-bandit", "hold-at-position", or "commence-approach").

Upon selecting a template, the user sees the current text of the message that will be sent, together with a set of menu buttons and entry boxes that allow the user to select (or type) pieces of information into the template. For example, most messages include the call-sign of the recipient of the message. The "hold-at-position"

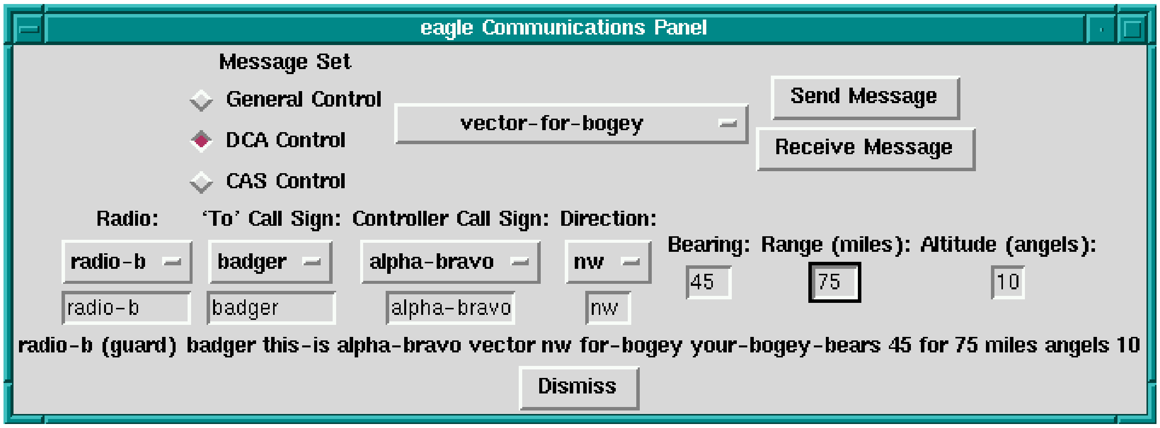

Figure 2 Communications Panel

message might additionally specify a bearing and range from an anchor point, and a holding altitude. After filling out the template with specific information, the user simply clicks "send", passing a string of English words to one or more agents. The agents then do the work of broadcasting the message more broadly, when appropriate. Figure 2 shows an example message being constructed using the CP.

Radio log

The companion tool to the CP is the Radio Log (RL). The RL tool can be attached to any individual agent, and it provides a running history of the English-language messages that agent has received or generated. Note that receiving a message simply means that the message was transmitted on a radio frequency the agent is tuned to. It does not necessarily mean that the agent was the intended recipient.

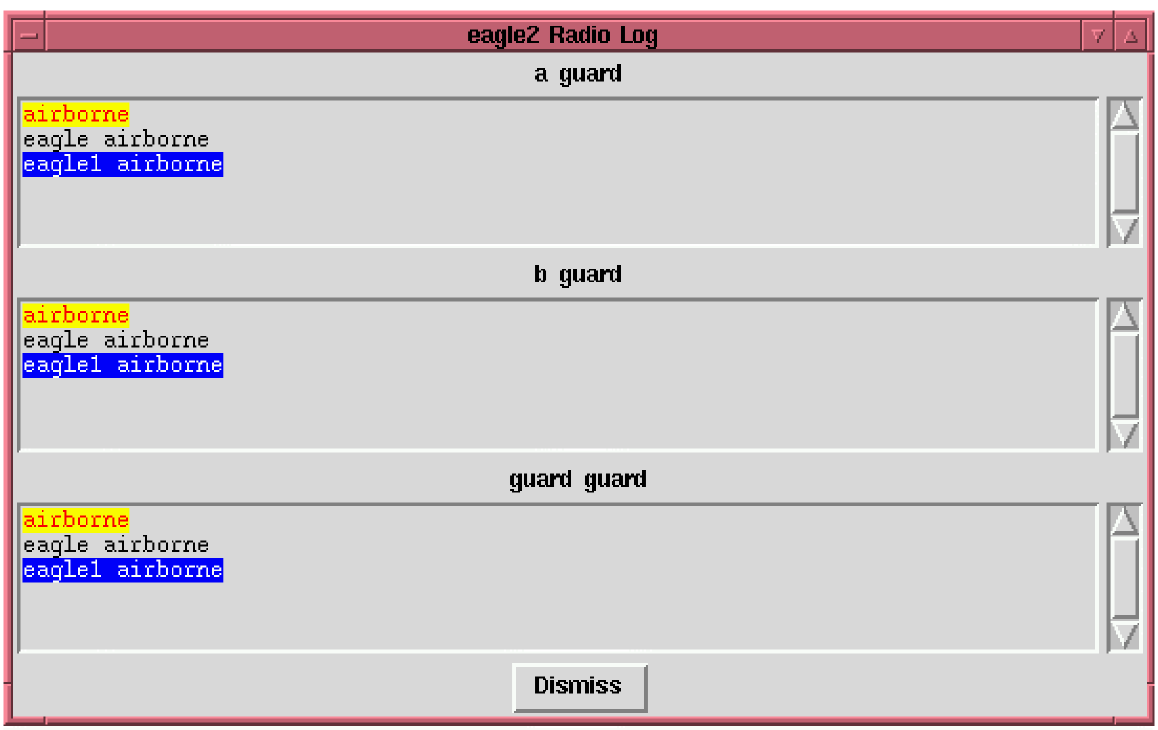

Figure 3 The Radio Log

The RL provides one text screen for each radio frequency that the agent is tuned to. Each text screen includes the entire history of messages for that frequency, from the time the RL tool was invoked. The messages are color- coded to distinguish between received and sent messages, and to highlight the most recent message on each frequency. Because the text strings match the language templates used by TacAir-Soar, each message is an easily readable, natural-language message. The messages are simply presented graphically, rather than run through a voice synthesizer, as with SoarSpeak. The RL not only complements the CP in providing communication between humans and synthetic agents, it also provides a valuable tool for observing the language interactions.

RADIOS

Arguably, the most common form of communication in military training is voice through radios. As stated earlier, TacAir-Soar agents use an underlying simulation system for modeling physical aspects of the simulated environment. This includes radios and effects on radio wave propagation. In ModSAF and its descendent JSAF [references], libraries provide a simulation of radios. These radios use DIS PDUs or HLA interactions for transmissions. Each radio can be tuned to a range of frequencies by the agent. Each vehicle used by Soar agents has a number of radios, the same number as in a real vehicle of that type (generally three in fixed-wing tactical aircraft). Both text strings and CCSIL may be communicated through the simulated radios.

Using radios as described provides a number of advantages. First, communication between all agents is handled in the same way regardless of their location in the network. Two agents communicate in the same way regardless of whether they are in the same process or in two separate processes on two separate computers. Second, effects such as jamming, spoofing, and interception can be easily simulated by injecting packets into the network's datastream or monitoring the datastream. Third, the format of the text messages facilitates matching messages against templates by agents. Messages are a list of constant strings and variables. Humans can easily read the text to understand what the agents are communicating. However, together with the ability to simulate distorted messages comes the requirement for CGFs to deal intelligently with such communication failures. Such capabilities in TacAir-Soar are still in early stages of development.

DATA COMMUNICATION

Not all explicit communication is speech. Military use of data communication is becoming more prevalent. For example, IFF transponders transmit information about an aircraft s identity using specific frequencies and formats. This section discusses two other types of data communication.

Status Reports

Status reports are a specialized type of communication content in a well structured format. These reports communicate a number of pieces of information to controllers. In the tactical air domain these include mission phase (proceeding to station, returning to base, etc.), time remaining on station, and munitions available. Status reports are sent out whenever the phase of the mission changes and at regular intervals during a phase (e.g. every 5 minutes while on station). The phases include proceeding to station, on station, attacking, refueling, and returning to base. All messages have some information in common, while other parts are dependent upon the phase. For instance, mission priority and weapons available are in all status reports, the station is reported while proceeding to a station or on station, and time remaining on station is only reported while on station

Data Links

A communication method related to radios is data links. They use radio frequencies in various ways but transmit various forms of data rather than a human voice. The information conveyed through the data link can be presented to the agents in a number of ways: text, coordinates, markings on a tactical data display, etc. As with voice radios, data links could be subjected to the same jamming and spoofing effects if required by exercises.

COMMUNICATION ISSUES

Though the major focus in this paper has been on the communication infrastructure and making communications understandable by humans, CGFs, and command and control systems there are a number of additional communication issues that bear mentioning here.

When to Communicate

One of the key components of effective communication is knowing when to speak and when not to. Observational results, confirmed by pilot experience, indicate that managing communication while involved in other intensive tasks, such as air to air engagements, can lead to cognitive overload. The general rule is communication by exception.

In some cases, military doctrine provides explicit directives for radio silence. For example, **Judy** is the command used by an interceptor aircraft to indicate the controller should stop providing positional information.

There is also an order to communication called comm cadence . This is a prescribed sequence that specifies whose turn it is to talk next, so that several entities do not attempt to speak at once.

Confirmation

In the fog of battle messages may be lost or corrupted and opponents may try to misinform. When a directive may be lost in transmission, we confirm receipt by saying **roger** and the sender checks to see if the confirmation is forthcoming. If not, the message is repeated. After several successive attempts, the entity it was directed to may be marked as unavailable.

In the case of possible radio spoofing we perform authentication. This occurs when a directive is given and the agents are unsure of the sender or when positional information is requested. Authentication may be handled either by a keyword based on the time of day or by an explicit request for authentication. The request consists of two keywords, and the response is a third. For example,

“Eyeball this is Eagle1, request you authenticate bravo charlie”

“Eagle1 I authenticate bravo charlie as foxtrot”

EXAMPLES

Having presented the varieties of communication facilities for TacAir-Soar, we conclude with examples of their use. The following situations each come from different applications of TacAir-Soar agents in training situations. Each highlights various aspects of communication in TacAir-Soar.

Virtual Control of Constructive Synthetic Forces

The intent of this scenario was to demonstrate the capability for human weapons controllers, in a virtual AWACS simulator at TACCSF, to direct and control intelligent CGFs in air-to-air combat as part of the COYOTE demonstration at AFRL, Mesa [Nielsen, et al., 2000]

This was a continuously fed air battle. As aircraft were lost, the controllers were free to scramble additional forces on ground alert, and as red forces were destroyed, new threats were generated. In order for the interaction to be as realistic as possible we used commercial off-the-shelf software to convert the controller’s voice directives to text messages, which were then broadcast over simulated radios. Responses were passed along these same radios and synthesized to speech, so the controllers could hear the responses over their headsets.

Once the controllers were trained in the vocabulary, the TacAir-Soar aircraft were able to carry out the controller’s orders directly without any intervention from an operator. Though TacAir-Soar is unable to understand arbitrary directives, it demonstrated a wide range of directives and control of the aircraft.

Synthetic Wingman

As part of the COYOTE demonstration at AFRL Mesa, a TacAir-Soar entity was tasked to fly formation with a human pilot in a virtual cockpit. Once again, SoarSpeak was used to convert speech directives to text and vice versa. This allowed the lead aircraft to commit against enemy aircraft, sort targets, and modify the tactical formation. TacAir-Soar maintained good formation with the virtual cockpit, performed independent targeting, and demonstrated close coordination. TacAir-Soar performed sufficiently well that the operators took turns rotating new pilots through the cockpits to test the capability of SoarSpeak operating in a high noise environment, and to account for individual speech differences. With literally ten minutes training in the vocabulary, a new pilot was able to lead TacAir-Soar-controlled aircraft into combat.

To illustrate the type of communication interaction used, when one pilot was attempting to "control" his wingman, his lexicon was less than precise. Consequently, the TacAir-Soar wingman moved out of visual range of his lead aircraft. When the flight lead (the human pilot) directed it to turn to heading 270, the TacAir-Soar aircraft responded "Roger, authenticate XYZ" while maintaining his current vector. The TacAir-Soar wingman was complying with theatre procedures that required him to verify unknown directives with coded authentication procedures. When the lead pilot authenticated accurately, the TacAir-Soar wingman immediately followed the lead's directions, and successfully rejoined the lead aircraft.

Air Traffic Control

In 1999, TacAir-Soar was incorporated into the Battle Force Tactical Trainer Air Management Node for the US Navy. This has been fielded on the multi-purpose amphibious assault ship LHD6, the *Bonhomme Richard*, in order to train air controllers performing air traffic and air intercept duties. The primary requirements for the CGFs are that they must behave appropriately and autonomously without control, but also respond correctly to direction from the controllers. [Jones and Kenny, 2000]

Rather than employ SoarSpeak, the Navy decided to employ a pseudo-pilot (human instructor) who accepts commands from the trainee and selects the appropriate phrase on a graphical Command Panel, to relay the directive to the TacAir-Soar aircraft. This allows much less precision in the trainee s vocabulary. Vocabulary and underlying behaviors to respond to the trainee s directives allow controllers to change course heading, speed, angle of approach, altitude, marshal points, and much more.

CONCLUSIONS

This paper discusses the difficult problem of communication in the simulated military aviation domain. As with most issues in training simulation, the problem needs to be approached realistically, interactively, and flexibly. We have demonstrated a suite of methods that work together for performing communications among our own CGFs, between our CGFs and humans, as well as between disparate types of CGFs. We hope that our multi-faceted approach to communication will provide the flexibility to make CGFs maximally useful in a variety of application areas.

REFERENCES

IBM (2000). IBM Software: Voice Systems. http://www-4.ibm.com/software/speech/

Jones, R. M. (1998). A graphical user interface for human control of intelligent synthetic forces. In *Proceedings of the Seventh Conference on Computer Generated Forces and Behavioral Representation.* Orlando, FL.

Jones, R. M., Laird, J. E., & Nielsen, P. E. (1998). Lessons learned from TacAir-Soar in STOW-97. In *Proceedings of the Seventh Conference on Computer Generated Forces and Behavioral Representation.* Orlando, FL.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine, 20*(1), 27-41.

Jones, R.M. and Kenny, P.G. (2000). Lessons learned from integrating STOW technology into an operational naval environment. In *Proceedings of the Spring SIW Conference.* Orlando, Fl.

Laird, J.E.. Johnson, W.L., Jones, R.M., Koss, F.V., Lehman, J.F., Nielsen, P.E., Rosenbloom, P.S., Rubinoff, R., Schwamb, K., Tambe, M., Van Dyke, J., van Lent, M, & Wray, R.E. (1995). "Simulated Intelligent Forces for Air: The SOAR/IFOR Project 1995." In *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation,* Orlando, FL.

Laird, J. E., Jones, R. M., & Nielsen, P. E. (1998). Knowledge-based multiagent coordination. Presence: Teleoperators and Virtual Environments, 7, 547-563.

Nielsen, P. E., Smoot, D., Martinez, R.,& Dennison, J. D. (2000). Participation of TacAir-Soar in RoadRunner and Coyote exercises at AFRL, Mesa, AZ. In *Proceedings of the Ninth Conference on Computer Generated Forces,* Orlando, FL.