Intelligent Automated Agents for Tactical Air Simulation: A Progress Report

Paul S. Rosenbloom,[1] W. Lewis Johnson,[1] Randolph M. Jones,[2] Frank Koss,[2] John E. Laird,[2] Jill Fain Lehman,[3] Robert Rubinoff,[3] Karl B. Schwamb,[1] and Milind Tambe[1]

[1]Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
{rosenblo, johnson, schwamb, tambe}@isi.edu

[2]Artificial Intelligence Laboratory
The University of Michigan
1101 Beal Ave.
Ann Arbor, MI 48109
{rjones, koss, laird}@eecs.umich.edu

[3]Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{jef, rubinoff}@cs.cmu.edu

Abstract

This article reports on recent progress in the development of TacAir-Soar, an intelligent automated agent for tactical air simulation. This includes progress in expanding the agent's coverage of the tactical air domain, progress in enhancing the quality of the agent's behavior, and progress in building an infrastructure for research and development in this area.

Introduction

At the Third Conference on Computer Generated Forces and Behavioral Representation we presented an initial report on an effort to build intelligent automated agents for tactical air simulation (Jones *et al*, 1993). The ultimate intent behind this effort is to develop automated pilots whose behavior in simulated battlefields is nearly indistinguishable from that of human pilots (and to go beyond this to develop generic agents that are readily specializable for this and other domains). If such agents can be created, they should provide close to ideal force supplements for many of the applications anticipated for distributed interactive battlefield simulation.

As of the initial report, prototype agents had been constructed that could, in real-time, flexibly use a small amount of tactical knowledge about two classes of one-versus-one (1-v-1) Beyond Visual Range (BVR) tactical air scenarios. In the *non-jinking bogey* scenarios, one plane (the non-jinking bogey) is unarmed and maintains a straight-and-level flight path. The other plane is armed with long-range radar-guided, medium-range radar-guided, and short-range infrared-guided missiles. Its task is to set up for a sequence of missile shots, at increasingly shorter ranges, until the non-jinking bogey is destroyed. Though such scenarios are not common in the real world, they are used as training exercises because they teach pilots how to position their planes for later shots while simultaneously taking earlier ones. In the *aggressive bogey* scenarios, one plane is attempting to protect a High-Value Unit (HVU), such as an aircraft carrier, via a Barrier Combat Air Patrol (BARCAP); that is, the plane patrols between the HVU and the anticipated threat (by cycling around a racetrack pattern), and then intercepts any threat that it detects in its sector. The other plane is attempting to attack the HVU, but to do so it must first intercept the defensive aircraft.

The prototype agents were all implemented as parameterized variations of a single multi-functional tactical-air agent, called *TacAir-Soar* (or *TAS* for short). TAS is built within *Soar*, a software architecture that is being developed as a basis for both an integrated intelligent system and a unified theory of human cognition (Rosenbloom, Laird, & Newell, 1993; Newell, 1990). Soar provides TAS with basic support for knowledge representation, problem solving, reactivity, external interaction, and learning. Soar also provides a potential means of integrating into TAS additional planning, learning, and natural language capabilities that are being developed independently within Soar.

The prototype TAS agents actually utilized only a subset of the capabilities provided either directly by Soar, or built separately within it. However, this subset – along with the domain-specific (and domain-independent) rules that were added to Soar's long-term memory – was sufficient to yield a combination of knowledge-based decision making, task(/goal) switching and decomposition, and real-time interaction with the DIS environment. Knowledge-based decision-making arises from Soar's ability to make decisions based on integrating *preferences* generated by arbitrary sets of rules. Task switching also arises from Soar's decision-making abilities, but here as specifically applied to the selection and switching of tasks. Tasks(/goals) are represented as *operators* in Soar, and are one of the main foci of its decision making. Task decomposition arises from using the same decision mechanism to drive task performance, plus Soar's ability to automatically generate a new performance context when a decision is problematic. When these mechanisms are combined with rules that generate preferences about which subtasks are appropriate for which problematic parent tasks (in the particular situation of interest), task decomposition occurs. Real-time interaction with the DIS environment arises from the combination of Soar's incorporation of perception and action within the inner loop of its decision making capabilities – thus allowing all decisions to be informed by the current situation (and interpretations of it, as generated by rule firings) – and the use of ModSAF (Calder *et al*, 1993) as the interface to the DIS environment (Schwamb, Koss, & Keirsey, 1994).

When combined with the very preliminary domain knowledge that was encoded at the time, this combination of capabilities yielded competent behavior for the non-jinking bogey scenarios, but only fragments of behavior for the aggressive bogey scenarios (due to insufficient knowledge about this class of scenarios). One type of aircraft, similar to an F14, was flown in these scenarios.

The purpose of this article is to provide a report, one year later, on the progress in moving TAS from the initial prototype agents towards the ultimate goal of human-like automated pilots that are broadly capable in tactical air scenarios. This report is intended to be complemented by the more detailed articles about particular aspects of TAS that also appear in these proceedings (Johnson, 1994a; Jones & Laird, 1994; Jones *et al*, 1994; Koss & Lehman, 1994; Laird & Jones, 1994; Rubinoff & Lehman, 1994; Schwamb, Koss, & Keirsey, 1994; Tambe & Rosenbloom, 1994; van Lent & Wray, 1994), rather than to substitute for them. Thus, where there is a potential overlap between this report and any of the more detailed articles, this report will become more terse and defer (and refer) to the appropriate detailed article(s).

In the body of this report, progress on *domain capabilities* will be covered first. The focus here is on expanding the classes of domain scenarios in which the agents can behave appropriately. Second, progress on *intelligent capabilities* will be covered. The focus here is on expanding the classes of basic intelligent abilities – such as coping with multiple interacting tasks, plan recognition, learning, planning, self-explanation, and natural language – exhibited by the agents. Third, progress on *infrastructure capabilities* – such as integration with the DIS simulation environment, low-cost interfaces for human pilots, knowledge acquisition, and documentation − will be covered. Finally, the article will be concluded with plans for the future.

Domain Capabilities

Progress on domain capabilities has occurred in two general areas: (1) improving the robustness and range of the scenarios within 1-v-1 BVR tactical air; and (2) scaling up the scenarios in terms of the number of vehicles, the range of vehicle types, and the complexity of the required organization and communication among the vehicles.

Within 1-v-1, the TAS agents can now exhibit competent behavior in the BVR tactical-air segments of the aggressive bogey scenarios. This includes the ability to patrol in a racetrack pattern; select radar modes, detect opponents on radar, perform search and acquire activities when opponents drop off of radar, and maneuver so as to confuse the opponent's search and acquire activities; determine and attempt to achieve appropriate intercept geometries and launch-acceptability regions (LARs); select, fire, and support missiles; and detect and evade enemy missiles.

As played out in the DIS environment, a typical aggressive bogey scenario involves an F14 which is defending its aircraft carrier against possible attack by a MiG29. The F14 patrols in a racetrack until it spots the MiG29 (the F14's radar and missiles both have longer ranges than do the MiG29's). The F14 continues to monitor the MiG29 until its commit criteria are achieved, at which point it begins the intercept by attempting to achieve a good geometry from which to fire a long-range missile (LRM). At some later point the MiG29 detects the F14 and also then begins an intercept. This makes it difficult for the F14 to achieve any further advantage in intercept geometry, so it gives up on that, and turns to maximize the rate of closure (and thus to minimize the time before the intercept is complete).

When the F14 is finally close enough (that is, it has the MiG29 within its LRM's *launch-acceptability region*), and is oriented correctly, it launches a long-range missile, and performs an *fpole* (a turn that decreases the rate of closure between the aircraft − to delay the arrival of any missiles that might have been launched from the MiG29 − while simultaneously keeping the MiG29 on the F14's radar). The MiG29 detects the fpole, and *beams* in response, by turning perpendicular to the F14 (to render blind the Doppler radar that is guiding the F14's missile). The F14 then attempts to search for and reacquire the MiG29, while simultaneously changing altitude in order to confuse the MiG29's search and acquire activities.

Both planes then generally attempt to set up for further missile launches, and to avoid missiles launched by their opponents. Depending on the exact timing of the engagement, and on the willingness of the two planes to take risks (this is a TAS parameter), zero, one, or both of the planes may be shot down in the process.[1]

This scenario can be played out with both planes flown by TAS agents, or with one or the other flown by a human pilot in a flight simulator. A formal demonstration of the aggressive bogey scenario in the WISSARD laboratory at Oceana Naval Air Station during June '93 successfully pitted two TAS agents in simulated F14s against two human pilots (in F18 simulators, but acting as MiG29s). This demonstration was set up as two independent 1-v-1 engagements (out of radar range of each other). Given the early state of development of the agents at the time, the human pilots were constrained in terms of the kinds of

---

[1]In real engagements, if one or more of the aircraft survive the BVR segment of the scenario, either a within-visual range (WVR) engagement − that is, a dogfight − or an air-to-ground attack on the HVU may then occur. However, these aspects of the scenario are not part of BVR tactical air, and are thus not pursued by the TAS agents.

tactics they were allowed to use. Under these circumstances the demonstration proceeded successfully, in real-time, and in an otherwise unscripted manner. The resulting behavior was much as described in the typical example above. Feedback from Navy personnel in attendance at the demonstration was uniformly positive.

Despite this demonstrable success – and the fact that in numerous subsequent presentations to domain experts and other Navy personnel TAS has consistently impressed with its quality of behavior – it must be noted that TAS is still not close to covering the full complexity of the domain abilities described above, or the interactions among them. For example, only a subset of the radar modes are used; search and acquire in three dimensions is not strong; and only a subset of the possible tactics for patrolling, confusing, intercepting, and evading are used. Fleshing out these abilities does not look conceptually difficult at this point, just time consuming.

Another dimension of complexity in 1-v-1 BVR tactical air that is not fully addressed at this point by TAS is the space of possible missions that the agents need to be able to perform. The aggressive bogey scenarios cover two types of missions (BARCAP-HVU and ATTACK-HVU); however, there is still a handful of others. One other mission to which TAS has recently been extended is a MiGSWEEP. A MiGSWEEP is a sweep by one side's fighters through the other side's territory to clear out a corridor for later aircraft (such as bombers). In addition to the abilities required for the previous missions, a MiGSWEEP requires the ability to fly to waypoints, and to break off an intercept and "blow through" an opponent (that is, engage in a small amount of WVR behavior in order to accomplish a high-speed pass of an opponent and continue with the planned flight path).

In scaling up from these 1-v-1 scenarios, the biggest change has been the incorporation of an ability to detect and manage multiple aircraft. In 2-v-1 engagements a *section* (i.e., a coordinated pair of planes) must be able to fly together in formation and execute coordinated tactics. In service of this they must be able to communicate with each other, and to be aware of each other's positions. The TAS agent is now capable of doing this (as discussed in the next section) to support competent 2-v-1 behavior, within the same kinds of limits described for 1-v-1 (Jones & Laird, 1994; Laird & Jones, 1994).

In 1-v-2 engagements a single aircraft must be able to identify and sort out the activities of a pair of adversaries who may or may not be flying together as a section (Jones & Laird, 1994). It must be able to work out intercept geometries that take both opponents into account – so as, for example, not to be sandwiched between them. It must also be able to determine which of the pair is the primary threat, target the primary threat, and determine when to also fire at the secondary threat. For example, if the pair are flying in a coordinated fashion, then firing a missile at one is likely to cause both to beam. It would thus be a waste to launch missiles at both under such circumstances. The current TAS agents are also capable of performing competently in such 1-v-2 engagements.

In 2-v-2 engagements, many of the same issues come up as in 1-v-2 and 2-v-1. However, additional capabilities are required to *sort* the opponents (determining which friendly aircraft has the responsibility for which opponent aircraft), to decide when one or both aircraft should launch missiles, and to decide when to split the single 2-v-2 engagement into two independent 1-v-1 engagements (i.e., to *strip*). Though work on 2-v-2 has just recently begun, there is now at least one working example of a section of TAS agents successfully sorting and firing at another section of TAS agents. Once 2-v-2 is completed, larger engagements (2-v-N, 4-v-4, N-v-N, etc.) will still remain to be covered.

Other aspects of scale up that are currently in progress include adding the ability to fly an F18 (to the original F14 and the recently added MiG29), and the addition of an air intercept control (AIC) agent in an E2 (a specialized radar plane that is similar to an AWACS) (Rubinoff & Lehman, 1994). The AIC's job is different in a number of ways from that of a fighter pilot, so stretching TAS to accommodate this new type of agent should force further generalization of its capabilities.

Intelligent Capabilities

With respect to intelligent capabilities, the most significant advance over the prototype agents has been the addition to TAS of the ability to maintain episodic memories of its engagements, and to use these memories in reconstructing what it did, why it did what it did, and what else it would have done if the situation had been slightly different (Johnson, 1994a; Johnson, 1994b). These description and explanation capabilities are available through an interactive debriefing interface, in which questions can be asked via selection from dynamically created menus, and answers are generated in (approximate) English. In contrast to explanation in most expert systems, where there is a distinct "explanation" system that has direct access to the "performance" system's knowledge and derivational traces, TAS generates the explanations itself based only on (1) what it can remember about what happened and (2) what it can later reconstruct about what it might have done (and why it might have done it). This is a process that can be misled by circumstances, but it is expected to be more like how human pilots would actually describe and explain their own behavior during post-mission debriefing (though the psychological and behavioral accuracy of this has not yet been studied).

In addition to these debriefing capabilities, significant progress has also been made on incorporating several other capabilities into TAS. One capability is coping with multiple interacting goals. Though mapping a forest of interacting goals onto the single goal stack maintained by Soar has turned out to be non-trivial − and is currently a topic of intensive investigation − workable strategies have been found for TAS agents to coordinate their behavior in the presence of all of these goals and their interactions (Jones *et al*, 1994). A second capability is integrating information from multiple sources about multiple agents (Jones & Laird, 1994). The sources of information about other agents have been expanded from just radar, to also include radio and vision;[2] and the number of agents about which information can be represented has been expanded from one up to an arbitrary number. A third capability is communication and coordination among multiple agents (Laird & Jones, 1994). Instead of modeling a group of related agents − such as a section of aircraft or a platoon of tanks − as a single aggregate unit, the behavior of groups is being modeled at the individual platform level. This provides additional flexibility and realism in the simulation, but also necessitates modeling how the groups actually do communicate and coordinate among themselves.

Additional capability investigations are also underway in the areas of learning, planning, plan recognition and natural language. Learning and planning are a relatively common part of Soar's repertoire of behaviors in general (Laird & Rosenbloom, 1990); however, they are not yet a routine part of TAS's behavior. Investigations of their use in TAS have begun − for example, the debriefing capability depends on learning being active within certain key portions of the TAS agents − but it is too early to comment generally on their outcome. In contrast, plan recognition is now a routine part of TAS's

[2]The radar, vision, and radio inputs attempt to provide TAS with the information a human would extract from those sources. However, this information is provided symbolically, and no actual visual or audio processing on the part of the agent is required.

behavior, but only of a simple, low-level, ad hoc variety. For example, when an opponent turns, a new (hand-coded) task may be selected to interpret whether the opponent is performing an fpole (as part of a missile launching plan) or a beam (as part of a missile evasion plan). General plan recognition turns out to be particularly difficult in the DIS environment because of the presence of partial information about multiple, flexible, interacting agents. However, a more systematic approach based on abstract *model tracing* (Anderson *et al*, 1990; Ward, 1991) in (multi-agent) world-centered models is being investigated in a version of TAS, and is showing some promise (Tambe & Rosenbloom, 1994). Finally, an investigation is in progress on how to incorporate independently developed, Soar-based, natural-language abilities (Lehman, Lewis, & Newell, 1991) into TAS (Rubinoff & Lehman, 1994). In theory, two automated agents could communicate without using natural language; however, to do so can affect how they are perceived by agents that are eavesdropping on them. In the longer run, natural language is also a critical capability if automated agents are ever to interact in a seamless way with human agents. Natural language communication will initially be provided between a pair of TAS agents − a fighter and an AIC (in an E2) − with further deployment hopefully to follow.

Infrastructure Capabilities

With respect to infrastructure, progress has been made on four topics: (1) integration of Soar with the DIS simulation environment; (2) provision of a low-cost interface for human pilots; (3) knowledge acquisition methodology; and (4) documentation tools and methodology. These topics are covered in turn here.

TAS agents are now able to act as full participants within the DIS battlefield simulation environment. The key to this has been the use of ModSAF 1.0 (Calder *et al*, 1993) as an intermediary between Soar and DIS (Schwamb, Koss, & Keirsey, 1994). ModSAF already contains an interface to DIS, so it was only necessary to add an interface between Soar and ModSAF. To do this we have implemented a *cockpit abstraction* on top of ModSAF that allows TAS to focus on behaving like a pilot, while ModSAF simulates vehicles, sensors, and weapons. TAS is not utilizing ModSAF's own pilot behaviors (such as Sweep, CAP, and Fly Route), as programmed into its tasks and task frames; however, TAS's piloting task has been simplified somewhat by providing it high-level flight control via a ModSAF library function that accepts as parameters a desired altitude, heading, etc. In addition to adding the cockpit abstraction (and getting Soar to use it), we have extended the implementation of Soar so as to allow multiple independent Soar agents within a single process. This has allowed multiple TAS agents to be compiled together with ModSAF in a single process,[3] and thus allowed communication between the agents and ModSAF to be mediated directly by calling library functions (rather than through slower interprocess communication mechanisms, such as sockets).

Given a cockpit abstraction, it turned out to be relatively easy to reuse it in support of a low-cost interface for human control of ModSAF aircraft. The Human Instrument Panel (HIP) provides an X-Windows-based interface to a vehicle's cockpit abstraction (van Lent & Wray, 1994). This enables a human pilot to perceive graphically-presented sensor information and to control the aircraft's flight, weapons, and sensors at the same level at which they are controlled by TAS agents. Easily being able to control ModSAF vehicles at this level of detail, and on any workstation, has proven to be quite useful in testing and experimenting with

[3]Soar is currently implemented in C − as is ModSAF − without which this integration would have been considerably more difficult.

TAS agents. However, the HIP clearly can't completely replace the functionality of higher fidelity (and cost) flight simulators.

With respect to knowledge acquisition, the most important development has been the opening of the WISSARD laboratory at Oceana Naval Air Station (in Norfolk, VA). The lab contains two high fidelity (dome) aircraft simulators; two medium fidelity aircraft simulators; plus workstations for running ModSAF, TAS, and several visualization and analysis tools. The laboratory has enabled us to add to the standard knowledge acquisition methodologies the ability to watch, tape, and log, engagements among human pilots (both official "subject matter experts", as well as operational pilots), and engagements between human pilots and TAS agents.

With respect to documentation, we have developed substantial portions of a three layer hypertext document that links together: (1) knowledge about the domain (as extracted from books, experts, etc.); (2) a description of the structure and content of TAS; and (3) the actual rules that comprise TAS (Koss & Lehman, 1994). This documentation has been developed within NCSA Mosaic, a distributed, multi-media, hypertext system. It is expected to facilitate understanding and validation of the knowledge and code embodied in the automated agents.

Summary and Future

TAS is now capable of performing competently in beyond-visual range tactical-air scenarios containing up to three interacting aircraft. Moreover, it can do so while flying two types of aircraft in service of three types of missions. It can also participate in interactive post-mission debriefings about its engagements.

These various capabilities arise from combining knowledge about the tactical air domain with a set of "intelligent" abilities embodied by TAS for knowledge-based decision making, reactive real-time interaction, coping with multiple interacting goals, integrating information from multiple sources about multiple agents, communication and coordination, episodic memory, and reconstructive self-description and self-explanation.

The basic TAS agent is coded within Soar via 145 operators, where each operator corresponds to a task (or goal) at some level of granularity. In terms of rules, the implementation involves approximately 1,500. Most of these rules are responsible for proposing, selecting, and applying the operators, but some do perform other tasks (such as encoding perceptual input, and elaborating state descriptions). The debriefing capability adds another 80 operators, amounting also to approximately 1,500 rules. So the combined system consists of 225 operators and approximately 3,000 rules. The natural language capabilities that are currently being added utilize an additional 56 operators, and approximately 900 rules. Note that these operator and rule counts are all "before learning", as learning can increase both the number of rules and the number of operators.

Beyond the agent itself, progress has also been made on building an infrastructure to support research and development on intelligent automated agents for tactical air, and beyond.

Plans for the coming year include completing 2v2 BVR tactical air, and transitioning TAS from tactical air to *close air support* (a form of air-to-ground engagement). We also expect to have planning, learning, and plan recognition working routinely in TAS, and to have limited amounts of natural language also in routine use. Meanwhile, incremental improvements are expected to continue on the infrastructure for research and development.

Acknowledgment

This research was supported under subcontract to the University of Southern California Information Sciences Institute from the University of Michigan, as part of contract N00014-92-K-2015 from the Advanced Systems Technology Office (ASTO) of the Advanced Research Projects Agency (ARPA) and the Naval Research Laboratory (NRL). Critical support has been provided by Dennis McBride of ARPA/ASTO; Tom Brandt, Bob Richards, and Ed Harvey of BMH Inc.; Andy Ceranowicz and Joshua Smith of Loral Inc.; and David Keirsey of the Hughes Aircraft Co.

References

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. 1990. Cognitive modeling and intelligent tutoring. *Artificial Intelligence* , *42*, 7-49.

Calder, R. B., Smith, J. E., Courtemanche, A. J., Mar, J. M. F., & Ceranowicz, A. Z. 1993. ModSAF behavior simulation and control. *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL: Institute for Simulation & Training. pp. 347-356.

Johnson, W. L. 1994. Agents that explain their own actions. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Johnson, W. L. 1994. Agents that learn to explain themselves. *Proceedings of the Twelfth National Conference on Artificial Intelligence*. Seattle: AAAI, In press.

Jones, R. M. & Laird, J. E. 1994. Multiple information sources and multiple participants: Managing situational awareness in an autonomous agent. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Jones, R. M., Tambe, M., Laird, J. E., & Rosenbloom, P. S. 1993. Intelligent automated agents for flight training simulators. *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL. pp. 33-42.

Jones, R. M., Laird, J. E., Tambe, M., & Rosenbloom, P. S. 1994. Generating behavior in response to interacting goals. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Koss, F. & Lehman, J. F. 1994. Knowledge acquisition and knowledge use in a distributed IFOR project. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Laird, J. E. & Jones, R. M. 1994. Coordinated behavior of computer generated forces in TacAir-Soar. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Laird, J. E., & Rosenbloom, P. S. 1990. Integrating execution, planning, and learning in Soar for external environments. *Proceedings of the Eighth National Conference on Artificial Intelligence*. Boston: AAAI, MIT Press. pp. 1022-1029.

Lehman, J. F., Lewis, R. L., & Newell, A. 1991. Integrating knowledge sources in language comprehension. *Proceedings of the Thirteenth Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ, Erlbaum. pp. 461-466.

Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Rosenbloom, P. S., Laird, J. E., & Newell, A. (Eds.) 1993. *The Soar Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press.

Rubinoff, R. & Lehman, J. F. 1994. Natural language processing in an IFOR pilot. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Schwamb, K. B., Koss, F. V., & Keirsey, D. 1994. Working with ModSAF: Interfaces for programs and users. *Proceedings of* *the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Tambe, M. & Rosenbloom, P. S. 1994. Event tracking in complex multi-agent environments. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

van Lent, M. & Wray, R. 1994. A very low cost system for direct human control of simulated vehicles. *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Ward, B. May 1991. *ET-Soar: Toward an ITS for Theory-Based Representations*. Doctoral dissertation, School of Computer Science, Carnegie Mellon University,

Biographies

Paul S. Rosenbloom is an associate professor of computer science at the University of Southern California and the acting deputy director of the Intelligent Systems Division at the Information Sciences Institute. He received his B.S. degree in mathematical sciences from Stanford University in 1976 and his M.S. and Ph.D. degrees in computer science from Carnegie-Mellon University in 1978 and 1983, respectively. His research centers on integrated intelligent systems (in particular, Soar), but also covers other areas such as machine learning, production systems, planning, and cognitive modeling. He is a Councillor of the AAAI and a past Chair of ACM SIGART.

W. Lewis Johnson is a project leader at the University of Southern California Information Sciences Institute, and a research assistant professor in the USC Department of Computer Science. Dr. Johnson received his A.B. degree in linguistics in 1978 from Princeton University, and his M.Phil. and Ph.D. degrees in computer science from Yale University in 1980 and 1985, respectively. He is interested in applying artificial intelligence techniques in the areas of computer-based training and software engineering. His current projects are developing tools that automate the generation of software documentation, and that explain the problem solving of intelligent agents.

Randolph M. Jones received his Ph.D. in information and computer science from the University of California, Irvine, in 1989. He is currently an assistant research scientist in the Artificial Intelligence Laboratory at the University of Michigan. His primary research interests lie in the areas of intelligent agents, problem solving, machine learning, and psychological modeling.

Frank V. Koss is a systems research programmer in the Artificial Intelligence Laboratory at the University of Michigan, where he is developing the interface between the Soar architecture and the ModSAF simulator. He received his BS in computer engineering from Carnegie Mellon University in 1991 and his MSE in computer science and engineering from the University of Michigan in 1993. He is a member of IEEE and AAAI.

John E. Laird is an associate professor of electrical engineering and computer science and the director of the Artificial Intelligence Laboratory at the University of Michigan. He received his B.S. degree in computer and communication sciences from the University of Michigan in 1975 and his M.S. and Ph.D. degrees in computer science from Carnegie Mellon University in 1978 and 1983, respectively. His interests are centered on creating integrated intelligent agents (using the Soar architecture), leading to research in problem solving, complex behavior representation, machine learning, cognitive modeling.

Jill Fain Lehman is a research computer scientist in Carnegie Mellon's School of Computer Science. She received her B.S. from Yale in 1981, and her M.S. and Ph.D. from Carnegie Mellon in 1987 and 1989, respectively. Her research interests span the area of natural language processing:

comprehension and generation, models of linguistic performance, and machine learning techniques for language acquisition. Her main project is NL-Soar, the natural language effort within the Soar project.

Robert Rubinoff is a postdoctoral research fellow in Carnegie Mellon's School of Computer Science. He received his B.A., M.S.E., and Ph.D. from the University of Pennsylvania in 1982, 1986, and 1992, respectively; his dissertation research was on ''Negotiation, Feedback, and Perspective within Natural Language Generation''. His research interests include natural language processing, knowledge representation, and reasoning. He is currently working on natural language generation within the Soar project.

Karl B. Schwamb is a Senior Programmer Analyst on the Soar Intelligent FORces project at the University of Southern California's Information Sciences Institute. He is primarily responsible for the maintenance of the Soar/ModSAF interface software described in this article. He received his M.S. in Computer Science from George Washington University.

Milind Tambe is a computer scientist at the Information Sciences Institute, University of Southern California (USC) and a research assistant professor with the computer science department at USC. He completed his undergraduate education in computer science from the Birla Institute of Technology and Science, Pilani, India in 1986. He received his Ph.D. in 1991 from the School of Computer Science at Carnegie Mellon University, where he continued as a research associate until 1993. His interests are in the areas of integrated AI systems, and efficiency and scalability of AI programs, especially rule-based systems.