Building Intelligent Pilots for Simulated Rotary Wing Aircraft

Milind Tambe, Karl Schwamb and Paul S. Rosenbloom
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
email: {tambe, schwamb, rosenbloom}@isi.edu

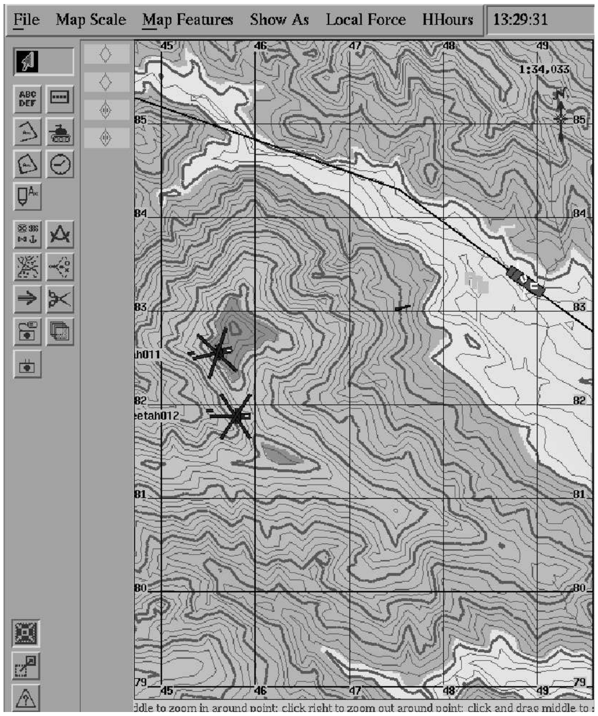
1. Abstract

The Soar/IFOR project has been developing intelligent pilot agents (henceforth IPs) for participation in simulated battlefield environments. While previously the project was mainly focused on IPs for fixed-wing aircraft (FWA), more recently, the project has also started developing IPs for rotary-wing aircraft (RWA). This paper presents a preliminary report on the development of IPs for RWA. It focuses on two important issues that arise in this development. The first is a requirement for reasoning about the terrain — when compared to an FWA IP, an RWA IP needs to fly much closer to the terrain and in general take advantage of the terrain for cover and concealment. The second issue relates to code and concept sharing between the FWA and RWA IPs. While sharing promises to cut down the development time for RWA IPs by taking advantage of our previous work for the FWA, it is not straightforward. The paper discusses the two issues in some detail and presents our initial resolutions of these issues.

2. Introduction

The Soar/IFOR project has been developing intelligent pilot agents (IPs) for simulated battlefield environments (Laird et al., 1995, Rosenbloom, et al., 1994, Tambe et al., 1995). Until Summer 1994, the project was focused on building IPs for simulated fixed-wing aircraft (FWA), including air-to-air fighters and ground-attack aircraft. Since July 1994, we have begun developing IPs for simulated rotary-wing aircraft (RWA), specifically, AH-64 Apache attack helicopters.

While there are similarities in an RWA and an FWA pilot's missions — e.g., employing weapons on targets, flying mission-specified routes — there are also some important differences. One key difference is reasoning about the terrain. For example, an RWA pilot's mission can involve flying Nap-of-the-earth (NOE), where it needs to fly only about 25 feet above ground level, while avoiding obstacles. It may also involve flying through a valley, or around a forested region. The mission may also involve hiding (masking) behind a ridge, popping up to spot enemy targets, and remasking in a new hiding position. Figure 1 provides an illustration of this type of terrain reasoning. It presents a snapshot, taken from ModSAF's plan-view display (Calder et al., 1993), of a typical scenario involving Soar-based RWA IPs. There are two RWA in the scenario, just behind the ridge, indicated by the contour lines. The other vehicles in the figure are a convoy of "enemy" ground vehicles — tanks and anti-aircraft vehicles — controlled by ModSAF. The RWA are approximately 2.5 miles from the convoy. The IPs have hidden their helicopters behind the ridge (their approximate hiding area is specified to them in advance). They unmask these helicopters by popping out from behind the ridge to launch missiles at the enemy vehicles, and quickly remask (hide) by dipping behind the ridge to survive retaliatory attacks. They subsequently change their hiding position to avoid predictability when they pop out later.



ddle to zoom in around point: click right to zoom out around point: click and drag middle to

Figure 1: A snapshot of ModSAF's simulation of an air-to-ground combat situation.

Thus, the development of RWA IPs brings up the novel issue of terrain reasoning, not addressed in previous work on Soar/IFOR agents. There has been much work on terrain reasoning in ModSAF in their development of semi-automated forces or SAFs

(Calder et al., 1993). That work has so far primarily focused on ground-based SAFs (e.g., (Longtin, 1994)), although there is a recent effort focused on terrain reasoning for RWA (Tan, 1995). Outside the arena of automated forces, terrain reasoning in the form of route planning and execution has been addressed extensively in AI and Robotics. The focus of much of this work is on 2D routes (Denton and Froeberg, 1984, Khatib, 1986, Lozano-Perez and Wesley, 1979, Mitchell, 1990) — and this category includes some previous work within Soar (Stobie et al., 1992) — although some efforts have also attacked the 3D route planning problem (Bose et al., 1987, Rao and Arkin, 1989). Other aspects of terrain reasoning such as tactical situation assessment (McDermott and Gelsey, 1987) and hiding (Stobie et al., 1992) have also received some attention, although not nearly as much as route planning. As discussed in Section 3, the pure route planning approaches from this literature are unlikely to address the terrain reasoning challenge facing the RWA IPs, which is to accomplish these tasks in real-time, given a realistic 3D terrain database. A hybrid solution combining some abstract plans with reactivity is currently being investigated.

Given the similarities between the FWA and RWA IPs, concept and code sharing between the two is a real possibility. Sharing would speed up development of RWA IPs by taking advantage of our previous work on FWA. However, the differences — such as the terrain reasoning capability above — imply that sharing is not straightforward. There have been some previous efforts aimed at facilitating reuse of code and concepts among Soar systems. These efforts have typically focused on reuse of individual capabilities, such as inductive learning (Rosenbloom and Aasman, 1990), or natural language (Lewis, 1993, Rubinoff and Lehman, 1994) capabilities. The novel issue here is that a large fraction of the FWA IP structure is potentially reusable in developing RWA IPs and such reuse needs to be facilitated.

The rest of this paper provides more details on these two issues. Section 3 focuses on terrain reasoning. Section 4 discusses the issue of code and concept sharing between Soar-based FWA and RWA IPs. We will assume some familiarity with the Soar architecture (Laird, Newell, and Rosenbloom, 1987, Rosenbloom, et al., 1991).

3. Terrain Reasoning

The overall terrain reasoning tasks for an RWA IP may be subdivided into two categories. The first is to fly from a given source to a destination, while abiding by mission specified constraints regarding the flight methods. A flight method primarily specifies maintenance of a certain air-speed and altitude above ground level. In particular, a *high-level* flight requires that the RWA fly more than 200 feet above ground level with air-speed as high as 145 knots. A *low-level* flight requires that the RWA fly 100-200 feet above ground level, while maintaining a maximum air-speed of 100 knots. A *contour* flight requires the RWA to fly between 25-100 feet above ground level, but with a maximum air-speed of 70 knots. An *NOE* flight requires the RWA to fly within just 25 feet above ground level, with a maximum air-speed of 40 knots. Additionally, an NOE flight may require that an RWA fly through a valley along a hillside, or through a narrow clear corridor in a forested region. The second category of terrain reasoning tasks involves an RWA IP's activities once it successfully follows its route to its battle area, and possibly engages enemy vehicles. Its activities in this area involve selecting and occupying good hiding positions (behind a ridge or a forested region) and flying between hiding positions while remaining concealed from a possibly mobile enemy. It may also involve reasoning about possible enemy hiding positions.

For both categories of tasks, one key issue for an RWA IP is to execute them in the context of a large-scale and realistic 3D terrain database, with features such as rivers, ridges, valleys, hills and forested regions. A second key issue is that given its complexity, the cost of sensing and processing large tracts of the terrain database is non-trivial. A third related issue is that an IP has to exhibit human-like behavior in performing these terrain reasoning tasks. Thus, it should not make use of information that a human pilot is unlikely to obtain. For example, as with a human pilot, an IP should plan routes using a map of the terrain database (which possibly may be inaccurate), rather than using the actual terrain database (which would always be 100% accurate). A final issue is that an IP has to perform its tasks in real-time. The following two subsections illustrate how these issues are addressed for each of the two types of tasks above.

3.1. Route Flying

For the task of route flying, one possible approach for addressing the above issues would be to use one of a variety of path-planning methods that provides a very detailed 3D point-to-point route, with little need or freedom to modify the given route (Stobie et al., 1992, Bose et al., 1987, Rao and Arkin, 1989, Denton and Froeberg, 1984). One such approach, based on weighted-region path planning (Mitchell, 1990), is to conceptually divide a map of the terrain into 3D cells (cubes), assign an appropriate cost to each cell that reflects mission-specified constraints, and then search for a minimum cost path through the cells. One advantage of such an approach is that an RWA IP need not sense the terrain database in any detail, but rather just enough to follow its plan. In addition, the low sensing overhead would facilitate real-time task performance. However, there are several problems with such an approach. First, given the complexity of the terrain, this approach would require a significant initial computational effort to create and then search the cells. Second, it could be wasteful given the realism of the RWA model and its flight controls — it will not be possible for a Soar-based IP to precisely control an RWA to follow such a detailed route, and it will end up having to reactively improvise the path or replan. The original planner could potentially take these realistic flight controls into account when developing a plan — so that no on-line replanning may be required — but that would further increase the complexity of planning. Third, if the map of the terrain is inaccurate or incomplete, the plan generated could be inaccurate as well. Even if the map were completely accurate (or if the IP were using the terrain database itself rather than a map), there could still be deviations from the planned route caused by an unexpected encounter with hostile or friendly vehicles. Thus, an IP may not be able to rely on just its original planned route; it may need to replan. Finally, human pilots typically do not rely on such detailed plans; and thus in forcing IPs to follow such plans, we are likely deviating from our goal of building human-like IPs.

So instead, a Soar-based IP follows a hybrid strategy that combines a plan-based and reactive strategy. In particular, it relies on more abstract route plans, that provide it just two to three intermediate points.[1] The IP then executes these route plans while reacting to sensory information that enables it to abide by the mission specified constraints. For ideal human-like IPs, this sensory information should be precisely what a human pilot would obtain visually by looking out the window. Unfortunately, for an IP, such visual processing is likely to be extremely complex and expensive. Therefore, special inexpensive sensors have been designed that approximate such visual processing. One such sensor is the *look-ahead altitude sensor* or *LAS* sensor. LAS is slaved to the parameters supplied by the IP. The IP sets parameters for LAS that specify a lookahead range and orientation, which in turn specifies a line segment of specific length and orientation originating from the IP's current location. Once these parameters are set, LAS scans the terrain database repeatedly (in fact, each time ModSAF schedules the agent for execution), and returns the highest altitude value along the specified line segment. For instance, to fly NOE, an IP sets LAS's parameters to a lookahead range of 50 meters, and orientation in the direction of its flight. The pilot reacts to LAS's response by modifying the altitude of its helicopter to be approximately 25 feet above the highest point.[2]

The top half of Figure 2 shows a pilot agent making use of LAS to fly NOE. The shaded portion in the figure is a profile of the terrain, while the dashed line is a profile of the helicopter flying NOE. The straight lines indicate LAS's lookahead range while scanning the database. The bottom half of Figure 2 indicates a longer lookahead range, and change in the flight profile that that results.

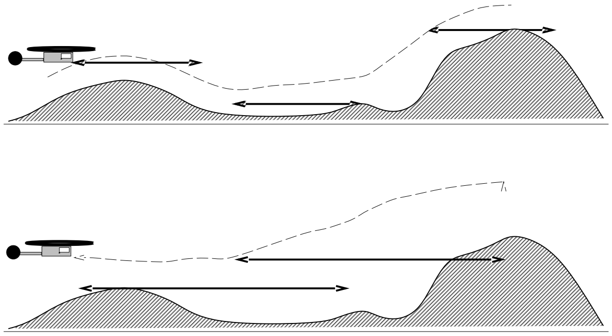


Figure 2: Illustrations of lookahead altitude sensor. LAS scans the terrain database each time the agent is scheduled for execution (illustrations not from an actual run).

The precise value of the lookahead range is determined to a large extent by the speed of the RWA. In particular, for an NOE flight, an IP currently flies conservatively at a speed of 20 knots. With 50 meters lookahead, that gives it about 5 seconds to change its altitude. The other flight methods, specifically contour, low-level and high-level flight, require that the RWA fly at a higher speed. This in turn requires that the IP set a longer lookahead range to give itself more time to react. Speed is however not the only factor determining the lookahead range. It is also dependent on the type of flight profile desired. For instance, at its speed of 80 knots, an IP could potentially sustain the altitude required for its low-level flight with a lookahead of just 200-300 meters. However, the flight profile generated follows the terrain much too closely — it is not as smooth as the flight profile that results from a human pilot's low-level flight (at least as indicated by the experts). Therefore, the low-level flight uses a much longer lookahead range of 1500 meters. The high-level flight uses a lookahead range of 5000 meters.

Unfortunately, long lookahead ranges in LAS could potentially hinder an IP's real-time performance. Therefore, to lower its cost, LAS samples precisely 100 points along the specified line

---

[1] At present, these abstract routes are provided by a human; although given that they are abstract, planning these routes is expected to be much less complex.

[2] RWA agents in ModSAF appear to follow a similar technique (Tan, 1995).

segment irrespective of the lookahead range. Thus, despite the variation in the lookahead range in Figure 2, LAS will scan precisely 100 points. This sampling resolution may appear to be very low, with the potential of missing high altitude cliffs. However, LAS's repeated scanning in effect improves its sampling resolution. In particular, since an RWA progresses towards its destination between two scans, successive scans sample slightly different points. In fact, each successive scan samples 99 points in the neighborhood of the points from its previous scan (on the same line segment), and one new point. This resolution could still be insufficient for some types terrain. For instance, if the terrain is an urban landscape with a sparse population of pin-shaped high-altitude structures,[3] there is a small possibility that LAS may miss those in its scanning. In such cases, there may be a need to increase the sampling resolution. However, the 100 point scans have so far proved adequate over the terrain database used in our experiments (the RWA have not crashed).

Figure 3 presents a flight profile from an actual run of a Soar-based RWA using the contour flight method. Figure 4 presents a flight profile from another run of a Soar-based RWA over approximately the same terrain, but using the NOE flight method. The shaded portion indicates the terrain, while the dashed line indicates the actual flight profile. IPs smoothen out the flight by using fuzz-boxes (McDermott and Davis, 1984) to avoid excessive altitude adjustments.

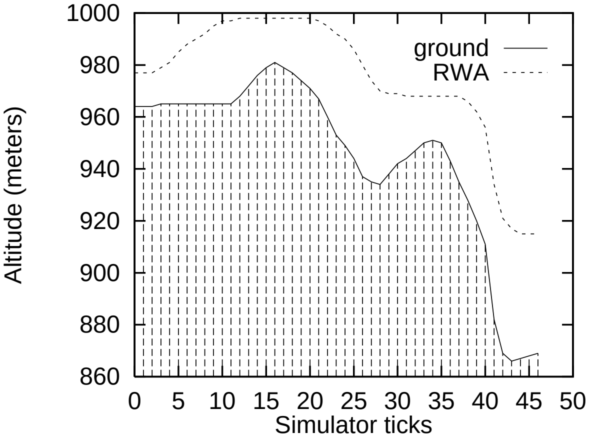


Figure 3: Illustration of a contour flight from an actual run.

Similar low-cost, LAS-type sensors approximating a human pilot's visual input are currently being designed to enable the RWA pilots to fly through valleys.

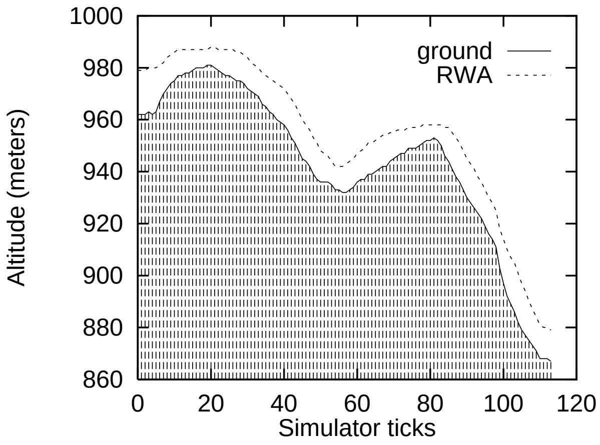


Figure 4: Illustration of an NOE flight from an actual run.

3.2. Hiding

Once an RWA IP reaches its mission-specified battle area, it needs to engage in hiding-related tasks. In general, a battle area could be of an arbitrary (convex) shape, or specified in terms of landmarks, such as trees or rocks. The IP should be capable of locating good hiding positions within this area and move between hiding positions while remaining concealed from its enemy. This second terrain reasoning capability, at least at this level of generality, is very much an issue for future research. At present, we have restricted the battle area to be a rectangle. One side of this rectangular area, typically coinciding with a ridge or a tree line, is a mission specified line segment. This is in essence considered to be an imaginary wall, and any movement behind it is assumed to be hidden from the enemy. An RWA IP hides in a small rectangular area (defined with a width of 100 meters) behind this imaginary wall. When relocating to a new hiding position, it uses the NOE flight method to remain at a low altitude and thus hidden behind the wall. The approximations of a wall and a rectangular area for hiding are both based on our previous work in the *groundworld* domain. Groundworld involved a simulated terrain with random configurations of horizontal and vertical walls, where an intelligent agent had to hide behind a wall to escape from another agent pursuing it (Stobie et al., 1992, Tambe and Rosenbloom, 1993).

4. Sharing and Reuse

RWA pilots' missions have some requirements — such as, identifying enemy vehicles, firing missiles at target vehicles and flying in formation — in common with those of FWA pilots. These commonalities may be exploited to cut down development time by sharing or reusing both code and concepts from Soar-based FWA pilots in the development of RWA pilots. For instance, for an FWA IP, the code for firing a

---

[3] A clock tower would be one example of such a structure.

missile involves three operators that orient its aircraft towards its target, then push a fire button to actually launch the missile, and then guide the missile (should the missile require guidance) via radar (or other) illumination of the target. These three operators can be reused in an RWA IP. At present, a Soar-based RWA IP has 44 operators, with 25 (that is about 57%) reused in some form from the Soar-based FWA IPs. The 19 new operators are those involved with terrain reasoning tasks such as flying NOE, masking and unmasking. This sharing is accomplished simply by loading in appropriate operators from an FWA IP code in an RWA IP.

Differences in concepts and terminology, however, make some of the sharing problematic. For example, for FWA pilots engaged in air-to-air missions, the concept of launch-acceptability-region or LAR of a missile combines both the range to a target and the target aspect (angle between the target's current heading and the straight line joining the target and the FWA pilot's current locations). Thus, if a target is heading towards the FWA pilot with a 0° target aspect, the missile may be fired from a long range; but the range is reduced substantially if the target has a 180° target aspect. In contrast, for an RWA pilot, the target aspect is irrelevant in calculating a missile's LAR — the missile may be fired at an equally long range irrespective of the target aspect. This creates a significant difference in the concept of a missile LAR for an FWA and an RWA IP, making the sharing of missile-LAR-related code difficult. There is an accompanying difference in the terminology as well — the RWA pilot refers to the missile LAR as a missile constraint.

At least some of these apparent discrepancies in the two IP's concepts — and potentially their terminology — could be resolved if the agents reason about the concepts from first principles. For instance, agents could calculate a missile's LAR from first principles, based on the relative velocities (speed and direction) of the missile and the target. Since an FWA IP's target in air-to-air combat is a fighter jet, moving at a speed that may be only a half to a fifth its missile speed, its angle of movement (target aspect) becomes an important factor in calculating LAR. In particular, a target moving towards the FWA allows a missile to be fired from a much longer range; while a target that is moving away requires that the missile be fired from a much closer range, so that the missile may catch up with the target before expending all its fuel. In contrast, an RWA IP's target is moving two orders of magnitude slower than its missile — the angle of the target's movement has a negligible impact on the missile range. In other words, with the first principles calculations, the target aspect discrepancy automatically disappears. It will appear important in FWA IP's calculations, and negligible in an RWA IP's calculations.

While such calculations from first principles would facilitate sharing, the calculations themselves may be prohibitively expensive, and hinder real-time performance. Soar's chunking (learning), could potentially compile such first principles calculations into new rules and alleviate this cost. However, that remains an issue for future work. We are currently relying on a lower cost alternative, where a problematic aspect of the agent code is rewritten when in reuse.

5. Current Status and Future Work

As of February 1995, the RWA agents are capable of performing a complete *attrit* mission, which involves flying to a battle area using one of the possible flight methods, followed by masking, unmasking, firing missiles at targets, and relocating to a different masking location between missile firings. We have run scenarios with up to four RWA IPs executing the *attrit* mission.

At present the RWA IPs can fly in coordination, in pairs. Extending this work to enable coordinated mission execution involving a platoon or a company of RWA agents (with a platoon and a company commander), is at the top of our agenda for future work. Agents at higher echelons of command, such as a company commander, may also bring up issues of communication and mission planning, which we have currently not addressed. Other issues for future work, mentioned in previous sections, include improvement in terrain reasoning for hiding, and in code/concept sharing among Soar agents.

6. Acknowledgements

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); and under contract N66001-95-C-6013 from the Advanced Systems Technology Office (ASTO) of the Advanced Research Projects Agency (ARPA) and the Naval Command and Ocean Surveillance Center, RDT&E division (NRAD). Critical expertise and support has been provided by David Sullivan of BMH Inc.

7. References

Bose, P. K., Meng, A. C-C., Rajnikanth, M. (1987) Planning flight paths in dynamic situations with incomplete knowledge. Proceedings of the SPIE conference on Spatial reasoning and multi-sensor fusion.

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 Conference on Computer Generated Forces and Behavioral Representation.

Denton, R. V., and Froeberg, P. L. (1984) Applications of Artificial Intelligence in Automated Route Planning. Proceedings of SPIE conference on applications of Artificial Intelligence. , pp. 126-132.

Khatib, O. (1986) "Real-time obstacle avoidance for manipulators and mobile robots". *International Journal of Robotics Research 5*, 1 , 90-98.

Laird, J. E., Johnson, W. L., Jones, R. M., Koss, F., Lehman, J. F., Nielsen, P. E., Rosenbloom, P. S., Rubinoff, R., Schwamb, K., Tambe, M., van Lent, M., and Wray, R., (May, 1995) Simulated Intelligent Forces for Air: The Soar/IFOR project 1995. Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation.

Laird, J. E., Newell, A. and Rosenbloom, P. S. (1987) "Soar: An architecture for general intelligence". *Artificial Intelligence 33*, 1 , 1-64.

Lewis, R. L. (1993) An architecturally-based theory of human sentence comprehension. Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.

Longtin, M. J. (1994) Cover and concealment in ModSAF. Proceedings of the Conference on Computer Generated Forces and Behavioral Representation.

Lozano-Perez, T. and Wesley M. A. (1979) "An algorithm for planning collision-free paths among polyhedral obstacles". *Communications of the ACM 22*, 10 , 560-570.

McDermott, D. and Davis, E. (1984) "Planning routes through uncertain territory". *Artificial Intelligence 22* , 107-156.

McDermott, D., and Gelsey, A. (1987) Terrain analysis for tactical situation assessment. Proceedings of the SPIE conference on Spatial reasoning and multi-sensor fusion.

Mitchell, J. S. B. (1990) Algorithmic approaches to optimal route planning. Proceedings of the SPIE conference on Mobile Robots.

Rao, T. M., and Arkin, R. C. (1989) 3D Path planning for flying/crawing robots. Proceedings of the SPIE conference on Mobile Robots.

Rosenbloom, P.S. and Aasman J. (August, 1990) Knowledge level and inductive uses of chunking (EBL). Proceedings of the National Conference on Artificial Intelligence. , pp. 821-827.

Rosenbloom, P. S., Laird, J. E., Newell, A., and McCarl, R. (1991) "A preliminary analysis of the Soar architecture as a basis for general intelligence". *Artificial Intelligence 47*, 1-3 , 289-325.

Rosenbloom, P., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Lehman, J. F., Rubinoff, R., Schwamb, K., and Tambe, M. (1994) Intelligent Automated Agents for Tactical Air Simulation: A Progress Report. Proceedings of the Conference on Computer Generated Forces and Behavioral Representation.

Rubinoff, R., and Lehman, J. (1994) Natural language processing in an IFOR pilot. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation.

Stobie, I., Tambe, M., and Rosenbloom, P. (Novemeber, 1992) Flexible integration of path-planning capabilities. Proceedings of the SPIE conference on Mobile Robots.

Tambe, M., and Rosenbloom, P. (July, 1993) On the Masking Effect. Proceedings of the National Conference on Artificial Intelligence.

Tambe, M., Johnson, W. L., Jones, R., Koss, F., Laird, J. E., Rosenbloom, P. S., and Schwamb, K. (Spring 1995) "Intelligent agents for interactive simulation environments". *AI Magazine 16* .

Tan, J. Flying NOE in ModSAF. Private communication.

8. Authors' Biographies

Milind Tambe is a research 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, India in 1986. He received his Ph.D. in computer science from Carnegie Mellon University in 1991. His interests are in the areas of integrated AI systems, agent modeling, plan recognition, and efficiency and scalability of AI programs, especially rule-based systems.

Karl Schwamb is a Programmer Analyst on the Soar Intelligent FORces project at the University of Southern California's Information Sciences Institute. He contributes to the maintenance of the Soar/ModSAF interface software and the Tcl/Tk interface to Soar. He received his M.S. in Computer Science from George Washington University.

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 and Fellow of the AAAI and a past Chair of ACM SIGART.

Table of Contents

1. Abstract
2. Introduction
3. Terrain Reasoning
    3.1. Route Flying
    3.2. Hiding
4. Sharing and Reuse
5. Current Status and Future Work
6. Acknowledgements
7. References
8. Authors' Biographies

List of Figures

Figure 1: **A snapshot of ModSAF's simulation of an air-to-ground combat situation.**
Figure 2: **Illustrations of lookahead altitude sensor. LAS scans the terrain database each time the agent is scheduled for execution (illustrations not from an actual run).**
Figure 3: **Illustration of a contour flight from an actual run.**
Figure 4: **Illustration of an NOE flight from an actual run.**