From: AAAI Technical Report FS-94-01. Compilation copyright © 1994, AAAI (www.aaai.org). All rights reserved.

Planning in the Tactical Air Domain*

Randolph M. Jones, Robert Wray, Michael van Lent, and John E. Laird
Artificial Intelligence Laboratory
University of Michigan
1101 Beal Avenue
Ann Arbor, MI 48109-2110
{rjones,wrayre,vanlent,laird}@eecs.umich.edu

Abstract

TacAir-Soar is a reactive system that uses recognition-driven problem solving to plan and generate behavior in the domain of tactical air combat simulation. Our current research efforts focus on integrating more deliberative planning and learning mechanisms into the system. This paper discusses characteristics of the domain that influence potential planning solutions, together with our approach for integrating reactive and deliberative planning.

TacAir-Soar (Jones *et al.* 1993; Rosenbloom *et al.* 1994) implements artificial, intelligent agents for use in tactical flight training simulators. The overall goal of the project is to create automatic agents that generate behavior as similar as possible to humans flying flight simulators. These agents will help provide relatively cheap and effective training for Navy pilots.

In order to accomplish this task, we need not only to acquire and encode a large amount of complex knowledge, but also to address a number of core research issues within artificial intelligence. Not the least of these issues is the ability for the agent to plan its activities appropriately, and to acquire efficient and effective new behaviors as a consequence of planning.

We are investigating the hypothesis that a variety of appropriate behaviors can arise from a system with a small, organized set of cognitive mechanisms as it interacts with a complex environment. Thus, the primary thrust of our research relies on *integration* in a number of different forms. Reactive behavior generation must be integrated with goal-directed reasoning and planning. These in turn must be integrated with other cognitive capabilities, such as situation interpretation, natural language understanding and generation, plan recognition, planning, etc. Rather than combining distinct modules for execution, planning, and learning, we are attempting to integrate all of these capabilities within a single control scheme. Thus, planning becomes simply another form of execution, which must interact with other knowledge in order to generate appropriate behavior. Learning occurs as a side effect of execution, manifesting itself in different ways depending on the particular tasks being executed. Because of the incremental, dynamic, and complex nature of behavior generation in the tactical air domain, learning must also be incremental, fast, and able to capture the complexities of goals and actions.

The current version of TacAir-Soar combines reactive and goal-driven reasoning to create what we call *recognition-driven problem solving* (Tambe *et al.* 1994). The system contains a large set of rules that fire as soon as their conditions are met, without search or conflict resolution. Some of these rules respond to immediate changes in sensory inputs, while others respond to higher-level interpretations of those changes and goals that the system posts for itself. As an example, TacAir-Soar may observe a series of readings about a contact on its radar, and conclude that the contact is an aggressive enemy aircraft. Thus, the system posts a goal of intercepting the aircraft, which involves maintaining a collision course. The actual heading of TacAir-Soar's aircraft will change every time the collision course changes. This paradigm for behavior generation is similar to *reactive planning* in the spirit of Firby's (1987) RAP planners. That is, the system does not perform any search to determine the best course of action, and it does not plan in terms of predicting future states of the environment.[1] It also computes its behavior dynamically, rather than generating a declarative plan that is later interpreted. Part of our current research effort is to equip TacAir-Soar with a deliberative planning component that separates

*Thanks to Paul Rosenbloom, Lewis Johnson, Soowon Lee, Frank Koss, and the reviewers for their comments on earlier drafts of this paper. In addition, this research has benefited enormously from the combined efforts of the Soar-IFOR group. The research is supported by contract N00014-02-K-2015 from the Advanced Systems Technology Office of the Advanced Research Projects Agency and the Naval Research Laboratory.

[1] TacAir-Soar agents do some prediction, but it is part of normal behavior generation, and not something that is learned about for decision making.

planning from normal execution by projecting future possible states and searching through them to decide on appropriate courses of action.

Of the following three sections, the first provides a short motivation for the usefulness of deliberative planning in the tactical air domain. The second lists a number of characteristics of the domain that have a significant impact on how planning must occur. These characteristics have been discussed in various earlier work on planning, but our work will address all of them together and attempt to provide a planning solution that naturally integrates into recognition-driven problem solving. The final section sketches potential solutions for deliberative planning. These solutions are suggested by a combination of the characteristics of the domain, our desire for a fully integrated system, and the problem-solving and learning paradigms provided by the Soar architecture.

Advantages of Deliberative Planning

As mentioned previously, the overall goal for the TacAir-Soar system is to generate human-like behavior within the simulated environment. One hallmark of human behavior is flexibility in the face of new situations. The current system has been equipped with a large knowledge base of tactics, vehicle dynamics, weapons characteristics, etc., and this allows the system to generate a wide variety of behaviors in response to different situations, missions, and goals. One approach to this type of domain has been to attempt to capture every possible situation that an agent may encounter in a recognition rule (e.g., Bimson *et al.* 1994). However, even if such an approach is possible, it would require extensive work on the knowledge base every time the domain changes a bit (for example, if new aircraft or missiles are developed).

In response to this difficulty, an agent must detect when it does not have suitable knowledge to react to a particular situation, and use its planning capabilities to generate appropriate actions based on more fundamental knowledge. This requires the agent to integrate deliberative planning with its current recognition-driven reasoning mechanisms. Naturally, we also expect the agent to learn from its planning episodes, generating new rules for future similar situations.

TacAir-Soar will do much of its planning "in the air," where there are tight restrictions on time, thus limiting the learning opportunities. However, human pilots often learn by flying real or simulated scenarios, and then debriefing the scenarios on the ground. By going back over each step of the scenario, the pilot can identify successes and failures, consider alternative courses of action, and take more time to evaluate various possible outcomes. Automated agents have also been demonstrated to benefit from such self-explanations (VanLehn, Jones, & Chi 1992). In addition, Johnson (1994a; 1994b) has presented a debriefing facility, in which TacAir-Soar agents can explain their actions after a scenario, and consider some hypothetical alternatives. The deliberative planning mechanism should expand on this approach and allow the system to learn from the debriefing experience. In addition, we intend the same planning mechanism to be used for planning both in the dynamic environment of an engagement and the calm, slow-paced environment of a debriefing session. Naturally, when the agent has more time to plan, the quality and quantity of effective learning should increase, but this will be due to the dynamics of the planning situation, not because of any differences in the planning and learning mechanisms.

Planning Issues for Tactical Flight

This section focuses on the specific aspects of the tactical air domain that have a significant impact on how planning should be carried out. There are five particular characteristics that set the domain apart from traditional domains used in planning research.

Interaction of Domain Goals

The current version of TacAir-Soar knows about almost 100 different types of goals, and many of these interact with each other. For example, there are times when an agent wants simultaneously to fly toward a target, evade an incoming missile, and maintain radar contact with another aircraft. This presents the traditional problem of planning for goal conjuncts (Chapman 1987; Covrigaru 1992). However, we must trade off the intensive search that can be involved in this type of planning with the dynamic and uncertain nature of the task (discussed below). Other researchers (e.g., Cohen *et al.* 1989; Veloso 1989) have suggested methods for planning about conjunctive goals in real time, and we hope to borrow from these approaches in our own efforts.

Two primary elements of conjunctive goal planning are detecting a goal interaction and then finding a way to deal with the interaction. Within TacAir-Soar, interactions will generally be detected when conflicting output commands are sent to the simulator (e.g., to come to two different headings) or when goal constraints are incompatible (e.g., turning away from a target while also maintaining a radar lock). In general, there will be two methods for dealing with such goal interactions. Some goals can be achieved conjunctively (perhaps not as efficiently as if the goals were independent), but sometimes it will be necessary to suspend certain goals temporarily when goals of higher priority (such as evading an incoming threat) conflict with them.

Dynamic, Real-Time Environment

As suggested above, TacAir-Soar cannot generally assume that it has ample time to plan. An agent may be planning an intercept course to a target when it detects an incoming missile. In this case, the agent must interrupt its planning in order to react in a timely

fashion. As a slightly different case, the situation may change so rapidly that the conditions that initiated planning may become obsolete before planning is completed. For example, the agent may begin planning which type of weapon it should employ against a target, only to find it destroyed by some other participant in the engagement. In both of these situations, the system should cease its planning activity, even if it did not find a result. Reactive planning systems (e.g., Agre & Chapman 1987; Firby 1987; Kaelbling 1986), and TacAir-Soar's recognition-driven problem solving address some of these issues by dynamically changing goals and behaviors as the environment changes. The next challenge is to integrate deliberative planning with dynamic reasoning in a smooth way.

Large State Representation

A further characteristic of the domain is that it involves rather large representations of the agent's current situation. The state representation includes information about various vehicle and weapon types, sensor information (from visual, radar, and radio sources), the agent's current mission goals, other "mental" annotations, and interpretations of the state, actions, and goals of other agents. For normal recognition-driven problem solving, the situated TacAir-Soar agent simply reacts to various features in this large state by generating actions or posting new goals or new interpretations of the situation.

The size of the state can impact deliberative planning in three ways. First, any time the agent wishes to plan, it must construct a copy of its current state representation. It can then manipulate this copy without changing its actual representation of the world or issuing real behaviors. Second, separating the two state representations allows the system to generate low-level reactions in response to one state while planning with the other. Because it takes some time to create this mental planning state, the agent should copy only the necessary information for planning and no more. Finally, some of the state information will be important to the current plan, while other information will be less important or totally irrelevant. It is not desirable for the agent to reason about portions of the state that have no bearing on the current decision. Thus, decisions about how much state to copy will have an impact on learning and the generality of new behaviors.

Planning in the Face of Uncertainty

A key feature of the tactical air domain is that there is generally a large number of participants in any given scenario. Some research (e.g., Georgeff 1984) has focused on this problem, and it naturally will have a strong effect on how TacAir-Soar can interpret and predict the consequences of its actions while planning. Anticipating the actions of cooperating agents may not be too difficult, because there exist social engagements and standard operating procedures between agents that cooperate. Predicting the future actions of competing agents is somewhat more difficult, and relies in part on recognizing the plans and goals of those agents (Tambe & Rosenbloom 1994; Wilensky 1981).

Given the unpredictable nature of modeling other agents, it is most appropriate for TacAir-Soar to create completable plans (Gervasio & DeJong 1994), in order to react appropriately to future actions by other agents. Contingency plans (Warren 1976) might also be useful, but these are generally expensive to generate. In a sense, TacAir-Soar's current knowledge base consists of a large completable plan, and such planning is consistent with our desire to integrate the current recognition-driven problem-solving structure with deliberative planning. The results of deliberative planning should be completable, reactive plans that the agent can execute and adapt in response to the dynamics of the environment.

Termination of Planning

As we have already mentioned, available time will have a large impact on how long any planning activity can continue. However, termination of planning is also influenced by when results can be produced. Most traditional planners have small sets of explicit, well-defined goals, and a precise evaluation function, so they can plan until a method is found to achieve their goals. Within the tactical air domain, there are many different types of goals, and different degrees to which they can be achieved. As an example, if an aircraft has the mission to protect its aircraft carrier, it may produce the goal of destroying an incoming attack aircraft. After the engagement has proceeded, the agent may find itself drifting from the carrier it is supposed to protect. At this point, it may decide that it has completed its mission by "scaring off" the threat, without actually destroying it, and it would be more dangerous to continue than to return to its patrol position.

The combination of limited reasoning time and ill-defined goals provides a further complexity for planning. The question is how far the planning process should continue, and when evaluation should take place.

Solutions for Deliberative Planning

These characteristics all have an impact on how planning can occur in an intelligent agent. Many of these issues have been addressed to some extent in previous research, but we hope to build an integrated system that addresses all of them. This section describes our preliminary efforts to develop an integrated planning solution that addresses all of the complexities of the domain. It begins with a discussion of the overall integrated framework, and then describes specific ideas for each of the planning issues.

Integrated Planning, Learning, and Execution

Our commitment to an integrated system began with our selection of the Soar architecture (Laird, Newell, & Rosenbloom 1987) as the platform for development. Soar provides an ideal basis for recognition-driven problem solving, and naturally supports the integration of execution, planning, and learning (Laird & Rosenbloom 1990).

Readers familiar with Soar will recall that all reasoning and behavior generation takes place in problem spaces, through the deliberate selection of operators. A fair amount of research on traditional planning within Soar (e.g., Lee 1994; Rosenbloom, Lee, & Unruh 1992) also organizes planning knowledge as sets of problem spaces. Problem spaces are collections of knowledge that address subgoals, which arise in response to a lack of knowledge in a particular situation. A typical example for planning occurs when an agent has a number of candidate actions to take, but does not have the knowledge to decide between them. For example, a pilot must decide which type of weapon to employ against a target, given the current mission and circumstances of the environment. After planning knowledge (e.g., a mental simulation of the alternatives) suggests an ordering, the automatic learning mechanism summarizes search in the problem space into individual rules ("chunks") that will apply in future similar situations.

We should stress the point that the natural representation for a plan within TacAir-Soar is not a declarative script of actions. Rather, a plan is a collection of recognition-driven rules and operators that apply opportunistically in response to particular patterns of sensor values, interpretations, and goals. Thus, in a sense, TacAir-Soar will never be learning entire plans, but it will be repairing or completing the general plan composed of all of its recognition rules.

Addressing Domain Issues

This integrated framework suggests possible solutions for planning that also address the issues presented earlier. To begin with, the high degree of interaction between goals suggests criteria for both triggering and evaluating new plans. Previously, we suggested that planning occurs when TacAir-Soar does not have the reactive knowledge necessary to choose between competing actions. This can be generalized to initiating planning any time the system detects an interaction between goals that it does not know how to handle. Covrigaru (1992) and Lee (1994) have investigated planning methods within Soar to address interactions between different types of goals. Evaluation of potential plans will be based on the resolution of individual interactions—as opposed to, for example, planning exhaustively until all interactions are resolved. As the agent develops responses to individual interactions, it can learn partial planning results in the form of new recognition rules.

These partial results also address the dynamic characteristics of the domain. Such planning will integrate smoothly with normal behavior generation because every planning episode will cause the system to learn *something*. If it is not something that completely resolves the current situation, it should at least allow the planning process to resume later without having to start over. Thus, particular planning efforts can be temporarily suspended (or perhaps abandoned entirely) without having been a total waste of time. When the system has ample time to plan (such as in a debriefing session), it is not clear whether the planning process will need to be qualitatively different. Presumably, the system will still be able to use its incremental planning techniques, but generate better quality plans because it has more time to evaluate and resolve interactions.

Also in response to to the dynamic domain, our initial efforts with TacAir-Soar have addressed the issue of integrating planning with execution. Many of the system's actions can apply without regard for whether the system is currently planning. For any aspects of the current situation that do not depend on the current planning activity, the system continues to generate behavior independent of other processing.

Because of TacAir-Soar's large state representation, we have adopted high-level, qualitative descriptions that summarize direct sensor readings, thereby reducing the amount of information that must be copied. In addition, the system attempts to make intelligent decisions about the portions of the state it cares about. These decisions are based on a static analysis of the domain knowledge, as well as dynamic reasoning based on the current situation. This allows the system to limit the amount of work it does in creating a mental copy of the state, which has been our primary concern in preliminary work on planning.

Our hope is that this approach will also aid the system in reasoning in an uncertain environment. As we have discussed, an appropriate response to this issue is to generate completable plans. In TacAir-Soar's terms, we wish to learn new rules for posting general goals, allowing the specific situation at execution time to dictate the precise actions that should be taken to satisfy those goals. Thus, a further aim for setting up a mental state for planning is to abstract away details that can be filled in by the situation later.

Finally, the criteria for terminating the planning process arise in part from the solutions we have already discussed. If there is time to plan exhaustively, the system will generate solutions for all the goal interactions it detects. Because the system returns incremental results as it plans, it is not as important for it to determine a fixed stopping criterion. If planning must be suspended temporarily, the partial planning results should allow planning to resume from where it left off. Finally, as we have mentioned, the system is able to generate behavior simultaneously with plan-

ning in many situations, so planning will not have to be interrupted until it is actually finished.

Summary

Simulated tactical air combat is an ideal, real domain for developing and testing new planning methods. The complexities of the task require us to focus on a number of planning issues that can be safely ignored in traditional planning domains. Although many of these issues have been addressed to some extent in the planning literature, we plan to provide an integrated solution to all of them. We have begun creating a system that smoothly integrates reactive and deliberative planning within the recognition-driven problem solving framework. Although our efforts with the deliberative planning component are young, our initial experiences have been encouraging. Hopefully, the complexities and real-time demands of the tactical air domain will lead us to a system that can model a continuum of planning processes from purely reactive to knowledge intensive and deliberate.

References

Agre, P. E., and Chapman, D. 1987. Pengi: An implementation of a theory of action. In *Proceedings of AAAI-87*, 123–154. Menlo Park, CA: AAAI Press.

Bimson, K.; Marsden, C.; McKenzie, F.; and Paz, N. 1994. Knowledge-based tactical decision making in the CCTT SAF prototype. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 293–303.

Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–377.

Cohen, P. R.; Greenberg, M. L.; Hart, D. M.; and Howe, A. E. 1989. Understanding the design requirements for agents in complex environments. *AI Magazine* 10(3):32–48.

Covrigaru, A. 1992. *Emergence of meta-level control in multi-tasking autonomous agents.* Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, University of Michigan.

Firby, R. J. 1987. An investigation into reactive planning in complex domains. In *Proceedings of AAAI-87*, 202–206. Menlo Park, CA: AAAI Press.

Georgeff, M. 1984. A theory of action for multiagent planning. In *Proceedings of AAAI-84*, 121–125. Menlo Park, CA: AAAI Press.

Gervasio, M. T., and DeJong, G. F. 1994. An incremental learning approach for completable planning. In *Machine Learning: Proceedings of the Eleventh National Conference*, 78–86. San Francisco, CA: Morgan Kaufmann.

Johnson, W. L. 1994a. Agents that explain their own actions. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 87–95.

Johnson, W. L. 1994b. Agents that learn to explain themselves. In *Proceedings of AAAI-94*, 1257–1263. Menlo Park, CA: AAAI Press.

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

Kaelbling, L. P. 1986. An architecture for intelligent reactive systems. In *Proceedings of the Workshop on Planning and Reasoning about Action*, 235–250.

Laird, J. E., and Rosenbloom, P. S. 1990. Integrating execution, planning, and learning in Soar for external environments. In *Proceedings of AAAI-90*, 1022–1029. Menlo Park, CA: AAAI Press.

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

Lee, S. 1994. *Multi-Method Planning.* Ph.D. Dissertation, Department of Computer Science, University of Southern California.

Rosenbloom, P. S.; Johnson, W. L.; Jones, R. M.; Koss, F.; Laird, J. E.; Lehman, J. F.; Rubinoff, R.; Schwamb, K. B.; and Tambe, M. 1994. Intelligent automated agents for tactical air simulation: A progress report. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 69–78.

Rosenbloom, P. S.; Lee, S.; and Unruh, A. 1992. Bias in planning and explanation-based learning. In Chipman, S., and Meyrowitz, A., eds., *Machine Learning: Induction, analogy and discovery.* Norwell, MA: Kluwer.

Tambe, M., and Rosenbloom, P. S. 1994. Event tracking in complex multi-agent environments. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation*, 473–484.

Tambe, M.; Jones, R. M.; Laird, J. E.; Rosenbloom, P. S.; Schwamb, K.; and Koss, F. 1994. Intelligent agents for interactive simulation environments. Manuscript in preparation.

VanLehn, K.; Jones, R. M.; and Chi, M. T. H. 1992. A model of the self-explanation effect. *Journal of the Learning Sciences* 2:1–59.

Veloso, M. M. 1989. Nonlinear problem solving using intelligent casual commitment. Technical Report CMU-CS-89-210, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Warren, D. H. D. 1976. Generating conditional plans and programs. In *Proceedings of the AISB Conference*, 344–354.

Wilensky, R. 1981. *Inside computer understanding: Five programs plus miniatures.* Hillsdale, NJ: Lawrence Erlbaum.