Adjustable Autonomy in Simulated Pilots

Nancy E. Reed and **Paul Scerri**
Real-time Systems Laboratory
Department of Computer and Information Science
Linköping University, S-581 83 Linköping, Sweden
nanre@ida.liu.se pausc@ida.liu.se

Abstract

We are currently developing autonomous agents to fly aircraft in simulation environments. In this paper we examine the requirements for automated pilots in virtual simulation environments and discuss where the ability to adjust the agent's autonomy level can help to either meet the requirements or improve the performance of the agent while meeting the requirements.

It is often desirable and sometimes necessary for the agent(s) in a simulated pilot to have adjustable autonomy. For example, simulated pilots need to react to situations in real time when it is not feasible to have a person in the loop. In other words, they need complete autonomy in some cases. It is also necessary in certain circumstances that a person be able to control or modify a simulated pilot's behavior to create a desired effect during a simulation. Under these circumstances, the simulated pilot's autonomy is reduced. Another common circumstance where adjustable autonomy is useful is when an simulated pilot must cooperate with other agents to accomplish some task, for example, maintaining a flying formation during a mission.

1 Introduction

Autonomous agents are being used in an increasingly broad range of applications ranging from space exploration to economic simulation. Agents must use data from their sensors to decide what actions (behaviors) to take using their effectors. The autonomy of an agent is characterized by the agent's ability to solve its goals without influence from the "outside". Outside influence can come from a person (external to the system) or from other agents (possibly within the same system).

We describe an ongoing project to build intelligent agents for a real-time simulation environment where varying levels of autonomy in the agents are useful. The agents must cooperate with each other under some circumstances and must allow a human controller to modify or influence particular aspects of an agents behavior while the agent continues to act. Challenging aspects of the problem include that the agents must continue to pursue one or more other tasks at the same time as accepting input from a human operator or cooperating with other agents on other task(s).

The rest of the paper is organized as follows. The next section defines autonomy and adjustable autonomy. Section 3 describes the domain of automated pilots for simulation environments and the following section describes a multi-agent representation for the simulated pilots. Section 5 discusses how the requirements for simulated pilots can be achieved or enhanced when the agents have adjustable autonomy.

2 Adjustable Autonomy

Agents may be completely autonomous, completely controlled (tele operated), or at some point in between – semi-autonomous [Kortenkamp *et al.*, 1999]. The degree of autonomy of an agent can be defined and measured with respect to one or more goals the agent is pursuing [Barber and Martin, 1999]. The degree of autonomy of an agent can change over time when its goals change or when a person (or another agent) takes more control or relinquishes some of its control on the agent's decision making process. An agent may be designed to always "follow orders" from a person or another agent, or it may have the ability to consider a person's or another agent's "suggestions" with varying levels of importance depending on the current situation.

Agents may cooperate with other agents, thus possibly "giving up" some of their autonomy to accomplish goals of mutual interest. In some situations, an agent may be completely unable to influence a decision (effectively having no autonomy for that decision). Suppose an agent votes for an action, but its vote is ignored by all others and a different action is performed.

3 Automated Pilots for Simulation Environments

Beyond visual range air combat involves highly complex military aircraft, fairly widely separated, relying mainly on radar for sensing and missiles to attack targets in the air or on the ground. Two systems for this domain are TACSI and TacAir-Soar. TACSI is a beyond visual range air-to-air tactics and combat simulator developed by Saab Corporation [Saab, 1998] and used for evaluating aircraft and flight tactics as well as for pilot training. Simulated pilots for TACSI have been developed and recent ones are described in [Coradeschi and Karlsson, 1996; Coradeschi, 1997].

TacAir-Soar [Jones *et al.*, 1993; Tambe *et al.*, 1995; Jones *et al.*, 1998] is a rule-based system, built within Soar, an architecture for general intelligence [Rosenbloom *et al.*, 1991]. The simulated pilots in TacAir-Soar use Soar to perform complex goal-directed reasoning and planning. A recent version of TacAir-Soar contains approximately 5200 production rules [Jones *et al.*, 1998]. TacAir-Soar has been used very successfully with a distributed interactive simulation (DIS) environment called ModSAF (Modular Semi-Automated Forces), funded by DARPA (Defense Advanced Research Projects Agency). In both TACSI and TacAir-Soar, human pilots may control simulated aircraft and interact with automated pilots during a simulation.

According to Tambe, et al. [1995], the requirements for automated pilots in virtual environments are: "(1) goal-driven behavior, (2) knowledge-intensive behavior, (3) reactivity, (4) real-time performance, (5) conformance to human reaction times and limitations, (6) overlap of performance of multiple high-level tasks, (7) multi-agent coordination, (8) communication, (9) agent modeling (especially opponent modeling), (10) temporal reasoning (dealing with time intervals), (11) planning, (12) maintenance of episodic memory, and (13) explanation." To conform to the requirements of the application, the automated pilot must produce a wide range of behaviors, including some that are very complex. For example, selecting the heading of the aircraft requires the combination of or negotiation of a number of different factors, potentially including avoiding any incoming missiles, maneuvering around mountains and other obstacles, reaching way points in the current path, achieving a position to attack targets, and maintaining spacing to fly in formation with team members.

4 A Multi-Agent Representation for Simulated Pilots

In simulated pilots previously developed for use with TACSI [Coradeschi and Karlsson, 1996; Coradeschi, 1997], pilot behavior was specified using hierarchical finite-state machines. The states represent the pilot's goals and her beliefs about the current state of the world. Rules produce the pilot's output and transitions between states. The rules each have priorities and the highest priority rule is executed if more than one rule is applicable.

We are using a multi-agent representation for simulated pilots. The agents in our simulated pilots are in a hierarchy where higher level agents contract agents at lower levels. At the lowest level of the hierarchy, agents negotiate with each other over an "output" of the simulated pilot (which goes to the environment simulator). At the start of a simulation, a small number of agents are created for each simulated pilot, one for each primary goal of the pilot. Example primary goals that automated pilots in TACSI might have are *intercept, escort, attack* and *maintain safety.* Other agents are created dynamically during runtime to accomplish the goals of the high level agents.

Communication between the automated pilots and TACSI are mostly at a low level, however some higher level instructions are defined in TACSI. Safeguards are built in to the aircraft controller in TACSI so that a pilot does not exceed the capabilities of the aircraft causing damage to it or themselves. If an instruction is given to the simulator that would violate the safeguards, it will not be performed.

5 Requirements and their Relationship to Autonomy

In this section we discuss how the requirements for automated pilots in virtual environments (presented in the same order as in Section 3) can be achieved or enhanced when the agents have adjustable levels of autonomy.

Requirement 1) *goal-driven behavior.* An agent's level of autonomy can be measured with respect to one or more goals. If the agent is completely autonomous, it will pursue the goal without outside interference. It may however not be able to successfully achieve the goal autonomously. When a person or external agent assists the agent in pursuing its goal(s), the agent's autonomy level is changed. Improving the chances of achieving a goal is a common reason to adjust an agent's autonomy. Adding or removing goals of the agent is another way of changing its autonomy.

2) *Knowledge-intensive behavior.* An agent uses knowledge of various types and from various sources in pursuing its goals. The agent's level of autonomy can be adjusted depending on how much of its knowledge is based on external agents or sources.

3) *Reactivity* and 4) *Real-time performance.* A simulated pilot must always react to its current situation in a "reasonable" amount of time. This puts constraints on how much time can be spent in deliberation. There may be less time for outside influences to reach the agent, affecting its autonomy level in meeting this requirement. Because humans are much slower than automated agents, an agent's fastest responses need to be performed without human intervention, there is not enough time for a person to react. An external agent can react with much greater speed and could be used to influence another agent, however. Humans should be able to influence and change the level of autonomy of an agent if desirable in the application when the reaction times are longer.

Modifications made to TacAir-Soar in order for it to be efficient enough to be used in real-time are described in Jones, et al. [1998]. These modifications included re-writing the Soar architecture in C, moving some of the computation out of the "reasoning" loop into the support code, and focusing attention by only performing computations necessary to the current system goals instead of performing calculations that could potentially be useful (geometrical, for example).

5) *Conformance to human reaction times and limitations.* Both upper and lower bounds on reaction times must be observed. Human-like behavior should also be produced. In other words, the simulated pilots should do things human pilots are able to do and might do. In contrast, a simulated pilot that flies in circles will completely lose credibility, even if that behavior might help in some small way. The need to simulate human reaction times puts constraints on how the simulated pilot can perform its goals. Achieving this requirement does not directly affect the agent's autonomy level.

6) *Overlap of performance of multiple high-level tasks.* To achieve multiple tasks in parallel, a simulated pilot must be able to reason about multiple tasks and possibly combine them. Each of the tasks or combinations of tasks may be performed most efficiently with a different level of autonomy. It could be useful to adjust an agent's level of autonomy with respect to its goal set to improve performance, however the integration of goals is usually handled by the agent autonomously.

7) *Multi-agent coordination* and 8) *communication.* Teamwork among agents is an example of coordination. In order to coordinate with other agents, an agent must be aware of the current situation and modify its behavior to best achieve the mutual goals. The agent's autonomy level may be changed by participation in a team.

Communication, when it comes from "the outside", may change an agent's level of autonomy. Communication is perhaps one of the most commonly used ways for multiple agents to coordinate behavior. For example, a team leader might order a group of pilots to follow him in a team formation. It is also possible to use sensing to try to determine another agent's role and performance within the team to effectively accomplish the team's goals without direct communication. An example would be performing a team mission under strict radio silence.

9) *Agent modeling* (especially opponent modeling), 10) *temporal reasoning* (dealing with time intervals) and 11) *planning.* A more accurate model of other agents or the world can increase the effectiveness of an agent's reasoning and planning to attain its goals. This could mean that the agent is more successful at accomplishing goals without needing assistance. In other words it can perform with a higher degree of autonomy in more situations, however the autonomy is not changed during execution.

12) *Maintenance of episodic memory.* Episodic memory can be used in the agent's planning. For example, an unsuccessful tactic might be repeated over and over again without a way to remember that it was not successful. The use of episodic memory could reduce the need for a human or another agent to modify the agent's behavior. Again, the presence of episodic memory does not directly affect the autonomy of the agent during execution.

13) *Explanation.* The ability of an agent to explain its actions is important for program development and improvement. The explanations can be used to improve the agent's capabilities in future situations. There appears to be no relationship between an agent's ability to provide explanations of its actions and its degree of autonomy.

Of these requirements for simulated pilots, those with the closest relationship to adjustable autonomy are goal-directed behavior, reactivity, real-time performance and multi-agent coordination.

6 Discussion

The multi-agent approach we are developing allows either human users or other agents to influence (and adjust the autonomy of) an agent. The influence can be at varying levels of abstraction and carry varying weights by creating or destroying agents that negotiate within the multi-agent hierarchies of the simulated pilots. The new agents would negotiate with other agents to produce an output that is a combination of "normal" agent behavior and user control.

By appropriately setting the priority of the newly created agents, the user's direction can override some or all of the other aspects of the agent's behavior. For example, if the aircraft should be forced to crash into the ground (for some specific reason), a new low-level high-priority agent should be created that will override other agents, i.e. those for avoiding the ground, in negotiations about the aircraft heading (output). In contrast, a task such as firing a missile on a particular ground position would require the creation of a high level, low priority agent that will not override agents such as those concerned with the safety of the aircraft.

Cremer [1995] describes a system using Hierarchical Communicating State Machines (HCSMs) which allows control of agents by other agents, however all communications are taken as orders in that system.

Effective tools to visualize the details of the multi-agent structure would be useful for a user to know how and where to change something to modify the agent's behavior in real-time. Even with an effective visualization tool it may be difficult for a person to control a complex agent or a large number of agents by creating and destroying single agents, demonstrating the need for interactions at multiple levels of abstraction.

A complementary approach is used in TacAir-Soar where humans are able to interact with the system and direct the simulated pilots using natural language and voice recognition software.

7 Summary

This paper describes the application domain of automated pilots for air combat simulations and discusses the requirements of the domain in relationship to the level of autonomy of the agents that comprise the simulated pilots. In different situations, it is desirable that the simulated pilots range from being completely autonomous to having little or no autonomy. The simulated pilot's autonomy might be changed by a human that directs or alters its behavior or by other agents through its interaction and cooperation with them in the simulation.

We are currently developing simulated pilots for TACSI. We are focusing on ways to specify pilot behavior that can be performed by non-programming experts, including pilots. A simulated pilot's behavior needs to be specified before a scenario with a relatively small (reasonable) amount of effort. In addition, there are situations where it would be very useful to modify a simulated pilot's behavior during a simulation - changing its autonomy.

Current work includes finishing the prototype implementation of the simulated pilots, interfacing the simulated pilots with TACSI and developing test scenarios and tactics for evaluation of the approach. Future work is planned to evaluate the ease of specifying simulated pilot (agent) behavior in the system by non-programmers, adding team behavior capabilities, and adding alternative interaction modes, for example graphical or natural language, for users to direct the simulated pilots.

8 Acknowledgments

This work is supported by Saab Corporation Operational Analysis Division, the Swedish Government under NUTEK grants IK1P-97-09677 and IK1P-98-06280, and by Linköping University under CENIIT grant 98.6.

References

[Barber and Martin, 1999] K. S. Barber and C. E. Martin. Agent autonomy: Specification, measurement, and dynamic adjustment. In Henry Hexmoor, editor, *Autonomy Control Software Workshop, Agents 99*, pages 8–15, May 1999.

[Coradeschi and Karlsson, 1996] Silvia Coradeschi and Lars Karlsson. Intelligent agents for aircraft combat simulation. In *Proceedings of the 6th Conference on Computer Generated Forces and Behavioral Representation*, Orlando, FL, 1996.

[Coradeschi, 1997] Silvia Coradeschi. A decision-mechanism for reactive and coordinated agents. Master's thesis, Department of Computer and Information Science, Linköping University, May 1997.

[Cremer *et al.*, 1995] James Cremer, Joseph Kearney, and Yiannis Papelis. HCSM: A framework for behavior and scenario control in virtual environments. *ACM Transactions on Modeling and Computer Simulation*, pages 242–267, 1995.

[Jones *et al.*, 1993] R.M. Jones, J.E. Laird, Milind Tambe, and Paul Rosenbloom. Intelligent automated agents for flight training simulators. In *Proceedings of the 3rd Conference on Computer Generated Forces and Behavioral Representation*, Orlando, FL, 1993.

[Jones *et al.*, 1998] Randolf Jones, John Laird, and Paul Nielsen. Automated intelligent pilots for combat flight simulation. In *Proceedings of the Tenth National Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pages 1047–1054, Menlo Park, California, 1998. American Association for Artificial Intelligence.

[Kortenkamp *et al.*, 1999] David Kortenkamp, Robert Burridge, R. Peter Bonasso, Debra Schreckenghost, and Mary Beth Hudson. An intelligent software architecture for semi-autonomous robot control. In Henry Hexmoor, editor, *Autonomy Control Software Workshop, Agents 99*, pages 36–43, May 1999.

[Rosenbloom *et al.*, 1991] Paul S. Rosenbloom, John E. Laird, Allen Newell, and Robert McCarl. A preliminary analysis of the Soar architecture as a basis for general intelligence. *Artificial Intelligence*, 47:289–325, 1991.

[Saab, 1998] Saab. The TACSI users guide. technical report TUCU-MI-95:103. Technical report, Saab Military Aircraft, 1998. Edition 5.2.

[Tambe *et al.*, 1995] Milind Tambe, W. Lewis Johnson, Randolph Jones, Frank Koss, John Laird, Paul Rosenbloom, and Karl Schwamb. Intelligent agents for interactive simulation environments. *AI Magazine*, 16(1):15–39, Spring 1995.