

Distributed Continual Planning for Unmanned Ground Vehicle Teams

Edmund H. Durfee

■ Some application domains highlight the importance of distributed continual planning concepts; coordinating teams of unmanned ground vehicles in dynamic environments is an example of such a domain. In this article, I illustrate the ideas in, and promises of, distributed continual planning by showing how acquiring and distributing operator intent among multiple semiautonomous vehicles supports ongoing, cooperative mission elaboration and revision.

An agent needs to be able to do distributed continual planning (DCP) if its application domain has at least three characteristics: First, the application should require that an agent's success depends on choosing immediate actions that flow into good choices for future options. It is this longer-term view that motivates the use of **planning** such that an agent should decide between alternative anticipated sequences of activities; otherwise, the application might be better served with simpler reactive agents that only decide on their very next actions.

Second, what the agent knows about the application domain, or what the agent's objectives are, or both, can change over time. Information about the domain could be revealed incrementally or could dynamically change in ways outside the agent's control, and thus, the agent should continually reevaluate its ongoing plans and revise or elaborate them to accommodate the changes. Even if its external environment does not change, the agent's goals or capabilities could evolve over time, similarly stimulating the need for **continual** planning.

Third, some of the dynamics in the agent's world are the result of the activities of other

agents, or some of the capabilities of an agent are available only through the intervention of other agents. Thus, the (continual) process of formulating plans can require the participation of multiple agents to combine their knowledge and expertise. Similarly, the process of executing plans could use the capabilities of multiple agents. If plan formation or execution requires multiple agents, the application requires **distributed** planning.

The deployment of multiple unmanned ground vehicles (UGVs) in hazardous environments, such as contaminated areas or battlefields, is an example of an application that has the characteristics that motivate the use of DCP. For example, the UGVs (figure 1) in a UGV team could work in concert to provide surveillance over an area. How they should distribute themselves (and, thus, their collective sensory resources) might depend on the continually evolving situation. Patterns of movement into and among positions must be planned carefully and coordinated to ensure consistent coverage of the areas to be surveyed.

In this article, I clarify the concepts of DCP by describing how our group (see Acknowledgments) has applied them to UGV teams, with specific attention to how we have implemented and used DCP for UGV teams engaged in scouting missions in a simulated military setting. This article summarizes particularly the DCP issues that arose in this application domain; a broader set of issues and experiments with real robots are reported elsewhere (Durfee, Kenny, and Kluge 1998).

Distributed Continual Planning Strategy

Consider a UGV team that is placed under the control of a human operator tasked with conducting a scouting mission in a hazardous



Figure 1. Unmanned Ground Vehicles.

environment. Assume that before the mission has begun, the operator and UGVs can easily communicate to formulate a plan, preferably in terms of a scouting mission (rather than in terms of robot commands). However, once the mission begins, communication between the operator and UGVs (and between the UGVs) is sporadic and uncertain. The application thus exhibits what Veloso and Stone (1998) call periodic time synchronization, where there are periods of time where communication is extremely reliable and coordination is not highly time constrained, interleaved with periods of time when communication and coordination are much harder.

These factors motivated us to develop UGVs, and operator interfaces, that internalize substantial military knowledge, including standard procedures for achieving objectives, and that know the current objectives of an operator. That is, when UGVs cannot rely on contact with the operator, they must have a sufficient model of the operator's intent to choose actions that still meet objectives in the continually changing world. We therefore treat a UGV as a semiautonomous agent that can sense its environment and continually elaborate and revise plans from its doctrine-based repertoire that are most suitable for achieving its assigned objectives given the circumstances.

The previous discussion emphasizes the need for continual planning. Because the UGVs are tasked in teams, the work also

incorporates distributed planning techniques for improving a UGV's awareness of the more global circumstances and coordinating the plans of UGVs. Because a human is part of the team, the UGVs treat the operator (or, alternately, the operator's workstation) as an agent that also is adopting plans (mission specifications) from its repertoire in response to input from the UGVs.

Rather than treat these loci of activity in a piecemeal fashion, this approach asserts that permeating all these tasks ranging from the specification of missions down to the invocation of robotic actions is knowledge about how to pursue these tasks that can be combined in flexible, situation-specific ways. Thus, this article describes how a procedural encoding of domain knowledge and mechanisms to use it provide a unifying DCP framework for (1) premission planning using distributed expertise and resources, (2) continual planning of doctrinally correct responses to unanticipated events, and (3) on-the-fly distributed coordination and replanning.

This framework has been realized by adapting procedural reasoning techniques (Georgeff and Lansky 1986), embodied in UMPRS (the University of Michigan procedural reasoning system [Lee et al. 1994]), to the planning and control of activities within, and across, UGVs and the operator. UMPRS controls a UGV based on its sensed situation and the directives and objectives handed down from the operator's workstation. UMPRS also controls the workstation by retrieving mission plans and taking actions to prompt the operator for refinements to objectives and mission parameters. Thus, the military intent and knowledge is distributed across the operator, the workstation, and the UGVs, and each of these can prompt another to allow distributed, mixed-initiative, and situation-specific definition, elaboration, planning, and revision of missions. The DCP capabilities spread among these various component agents is depicted in figure 2.

Premission Planning

Planning, including DCP, presupposes that agents have objectives that their plans are focused on achieving. A precursor to UGV teams engaging in DCP is that intent and objectives of (human) supervisors need to be conveyed to the UGV team. *Premission planning* is the opportunity for the system to ascertain this information by assisting the operator in specifying the mission goals and constraints as well as automating the translation and elaboration of military objectives into robotic actions.

At the heart of the system's premission planning is the UMPRS agent architecture, a C++ implementation of procedural reasoning concepts (Georgeff and Lansky 1986), tailored specifically to support the development of agent technologies for operator assistance and autonomous execution. UMPRS supports specification of a mission's operation order in terms of a sequence of military procedures, each performed within the context of a larger set of objectives and constraints. Thus, whereas the robotic plan executed directly on the vehicles consists of a sequence of primitive commands, it is within the UMPRS system that the actual intent of the mission, along with the expected (and possibly alternative) strategies for its accomplishment, are represented.

This more complete knowledge allows the UMPRS-based premission planner to assist the operator in decomposing the mission into more detailed segments, possibly formulating some portions of actual robotic plans, making suggestions to the operator about features of the mission plan, and prompting the operator for further input to resolve ambiguities in the specification. As it elaborates the mission, the planner can call on more specialized tools to formulate portions of the robotic plan. These include tools for route planning (Stentz 1995), planning observation points (Cook, Gmytrasiewicz, and Holder 1996; Kluge, Weymouth, and Smith 1994), and planning formations (Balch and Arkin 1998). The premission planner incorporates the information returned by these tools into its representation of the sequence of plan steps.

Because many plan details will depend on the circumstances in which the mission is being accomplished, the premission planner should only elaborate the plan steps to an intermediate level of detail. It should then pass the resulting sequence of subgoals (with constraints on their accomplishment) to the UGVs, which will each then elaborate their subplans further to develop the detailed robotic plan. This decomposition allows each UGV to continually revise and refine its own plan, within the larger outlines dictated by the global mission.

Finally, sometimes the circumstances that arise during execution can have been anticipated. Some foreseeable contingencies can be addressed completely locally by a UGV, such as what to do if an obstruction partially blocks the road. Others might include collective actions, such as prespecifying that a UGV should respond to an attack by notifying other UGVs and collectively falling back to a predefined position. By projecting proposed plans

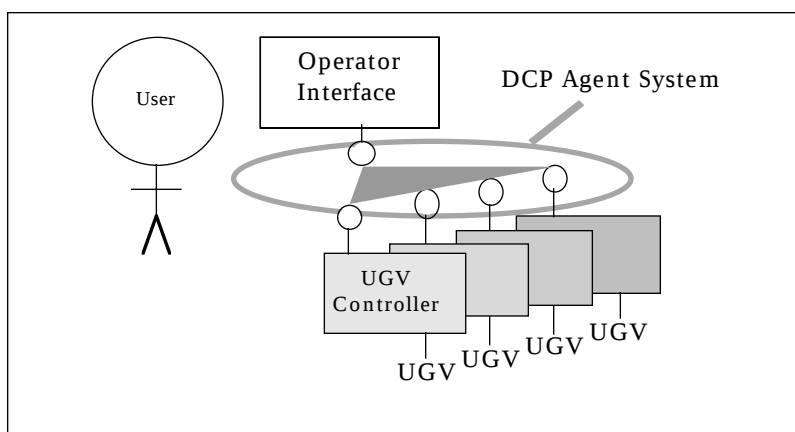


Figure 2. The Agent System Supports Distributed Continual Planning.

forward and modeling possible contingencies, the premission planner can consult the human operator before the mission about what to do (for example, where to fall back to) when contingencies arise and include the appropriate parameters in the UGV task specifications so that they react appropriately.

Continual Planning and Execution

Because this architecture assigns a UMPRS agent process to each UGV, a UGV elaborates plans both before and during a mission, such that it is up to each vehicle to decide how best to accomplish the objectives that it is handed from the operator's workstation. A clear advantage of this strategy is that the UMPRS mechanisms for plan elaboration are inherently responsive to changing circumstances. When deciding what primitive action to take next, a vehicle will consider alternative ways of accomplishing its next goal, taking into account the current context in which it is operating. As context changes, details of how goals are being accomplished will change (different procedures will be retrieved from the library of standard operating procedures), within the more persistent framework of higher-level procedures that are still being adhered to.

The context in which a vehicle operates includes its awareness of the external environment, the internal status, and the activities of other vehicles. Information from all these sources must be assimilated to classify the current operating conditions (Kluge, Weymouth, and Smith 1994). In this architecture, this assimilation is done through fusion processes that act on information stored in the information assimilation database (IADB). The IADB uses a CODGER-like blackboard (as developed for

autonomous land vehicles at Carnegie Mellon University [Stentz 1990]) to store the information about the world used by the UGV agent process. The information is collected from UGV sensors (for example, a camera), internal monitors (for example, a fuel gauge), and communications systems (for example, a radio) and combined into symbolic assessments of the situation (for example, *enemy-nearby?* is true). To establish context for invoking a procedure, the UMPRS process can query the IADB. To ensure that a procedure remains valid, the UMPRS process can pose a standing query to the IADB, such that the IADB will respond when the conditions of the query are met. Moreover, in some cases, a query from the UMPRS process can trigger the IADB to return goals back to UMPRS. For example, if UMPRS wants to know whether a route is safe, the IADB might trigger observation point planning, essentially saying that it can answer the question if the vehicle first performs a sequence of observations.

Finally, it is generally assumed that missions are being carried out by multiple UGVs, a critical component of context is the status of other friendly vehicles. Many cooperative procedures embed communicative actions for maintaining this context, such as in a *bounding overwatch* maneuver where two vehicles leapfrog through an area, taking turns watching over the other and then being watched over in turn, where the turn taking is often synchronized through messages. However, in cases where messages fail to materialize or where vehicles should maintain radio silence, the vehicles can use observations made with their sensors to draw inferences about the status of other vehicles (including enemy vehicles) (Huber, Durfee, and Wellman 1994).

Distributed Continual Planning

As more significant and unpredicted changes to circumstances accumulate, the UGVs continually examine and reformulate their local plans and might have to engage in distributed replanning. For example, if the vehicles were intended to perform a triangulated reconnaissance to localize some enemy force, the mission could be jeopardized if some vehicles become inoperative or lost. The continual planning activities occurring locally on UGVs need to work in concert to revise team plans, such as distributing the operational vehicles in radically new locations to approximate the localization effort, rapidly moving vehicles among multiple points so that each is making several observations, or aborting the initial objective and pursuing a secondary mission goal.

In this system, each UGV can use its own local IADB to update its knowledge about the situation. Unexpected changes trigger the retrieval of alternative plans for achieving the vehicle's goals. In addition, because a vehicle maintains top-level goals, including keeping other vehicles and the operator workstation informed of significant context changes, it volunteers some of the contents of its IADB to others. Consequently, their IADBs are updated, triggering further changes either in other vehicles or across the mission at the operator workstation. In fact, these top-level goals inherently are geared toward coordination and can trigger cooperative replanning among the vehicles, should they detect that their current plans are outdated and that the operator is unable (not responding) to oversee their collective replanning activity.

To ensure coordination, it is important that the range of replanning choices available to the vehicles be limited enough to avoid conflict, without being so limited as to unnecessarily constrain the vehicles. General constraints, such as so-called *social laws*, can serve this purpose (Shoham and Tennenholtz 1994) but can impose unnecessary restrictions on vehicles' activities given their current objectives and intentions. For example, a social law that makes each route one way to ensure no collisions occur can incur inefficiencies in a situation when a one-way road could safely be used in the opposite direction to reduce a vehicle's distance traveled. An alternative is to allow the vehicles to exchange explicit information about their revised plans and search for a minimal set of commitments that assure conflict-free execution (Clement and Durfee 1999; Lee 1996).

Implementation

Our group implemented the previously described component technologies to build the interacting agent mechanisms that provide DCP capabilities within the UGV application. UMPRS was the backbone of each of the agents, both for on-board vehicle planning and execution and control of the functions of the operator workstation (OWS). For mission planning, UMPRS was interfaced to planning tools for formation planning, route planning, and observation point planning. The IADB was implemented and interfaced to UMPRS.

A true evaluation of these technologies in a military setting would involve four (working) UGVs with reasonable sensing capabilities operating in an environment where some unexpected contingencies would arise. Lacking access to such a setting, we approximated

it using the MODSAF simulation system (Calder et al. 1993), a powerful, widely used simulator for military purposes.

In MODSAF, vehicles are controlled by finite-state machines (FSMs), where the user gives the vehicle a sequence of tasks, and the vehicle blindly executes them. For our purposes, we needed to have more control over the vehicle, so we built into the simulator the ability for the vehicle to receive commands from the UMPRS system, along with the ability to run some FSM tasks such as moving to specified points. In this way, we exploited legacy methods for directly controlling the vehicles' navigation, sensing, and communications. The setup of the system is summarized in figure 3, involving four vehicle agents and one OWS agent running UMPRS. The OWS agent was connected to the legacy graphic user interface (GUI) from Hughes STX. The IADB was integrated in, along with a route planner and an observation point planner (OPP). The knowledge possessed by the agents was fairly impoverished: The OWS was knowledgeable about only a few kinds of mission, and the UGVs had knowledge to achieve basic mobility and perception goals.

We tested the system in scenarios ranging from one to four UGVs. A typical scenario we used is as follows (figure 4): An operator wants to scout ahead across enemy lines, such that the scouts eventually take up observation posts overlooking a suspected enemy encampment. This goal would be selected from among the (few) options known to the mission planner; the planner retrieves an appropriate procedure for generating the mission by binding map measures to variables, prompting the operator for more data, and invoking other planning tools (such as deciding placements of vehicles during a bounding overwatch). The procedure is executed and the mission plan formed in terms of a series of subgoals associated with each vehicle. These subgoal sequences are sent to the different UGVs, which run their own UMPRS processes to elaborate these further into specific robotic behaviors (and, along with these, their associated preplanned contingency behaviors). These behaviors are downloaded to MODSAF, and the vehicles begin executing their plans.

In the course of moving to their ultimate destinations, vehicles eventually become aware of an unexpected enemy in a nearby location. The sensory uncertainty captured in MODSAF means that when they would detect the enemy would vary from run to run. On discovering the enemy, the vehicles alert each other and fall back to previous checkpoints.

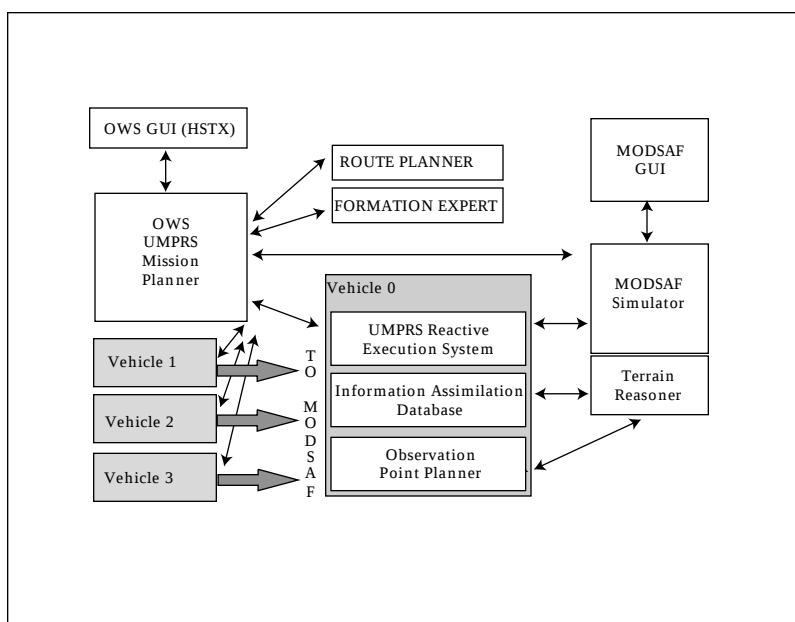


Figure 3. Demonstration Setup.

This is a predefined contingency response and so is executed quickly. As they fall back, however, the updated IADB is probed by the UMPRS process on board each vehicle. The UMPRS process on the vehicle that first spotted the enemy would feed the new information into the OPP, which returns new observation points for the vehicle such that it could better isolate the extent of the unexpected enemy presence. The vehicle then leaves its fallback point and carries out this revised mission (figure 5). Variations on this theme have the feedback going up to the IADB at the OWS and the OWS planning new observation points for several vehicles rather than having a single vehicle take on the task alone.

Lessons Learned

The strength of incorporating DCP concepts into UGV teams can be appreciated by comparing this system to the mission planning done with real UGVs during a demonstration led by Lockheed Martin. In the Lockheed Martin demonstration, the development of a mission plan for three vehicles could take several hours of two operators' time. The operators would micromanage the vehicles during execution. Sometimes they would get calls from the field requesting that a vehicle be inched forward a little to look over a ridge. Sometimes they would use views from the UGV camera to realize that the UGV was going to the wrong place and would quickly generate and download a new sequence of robotic actions.

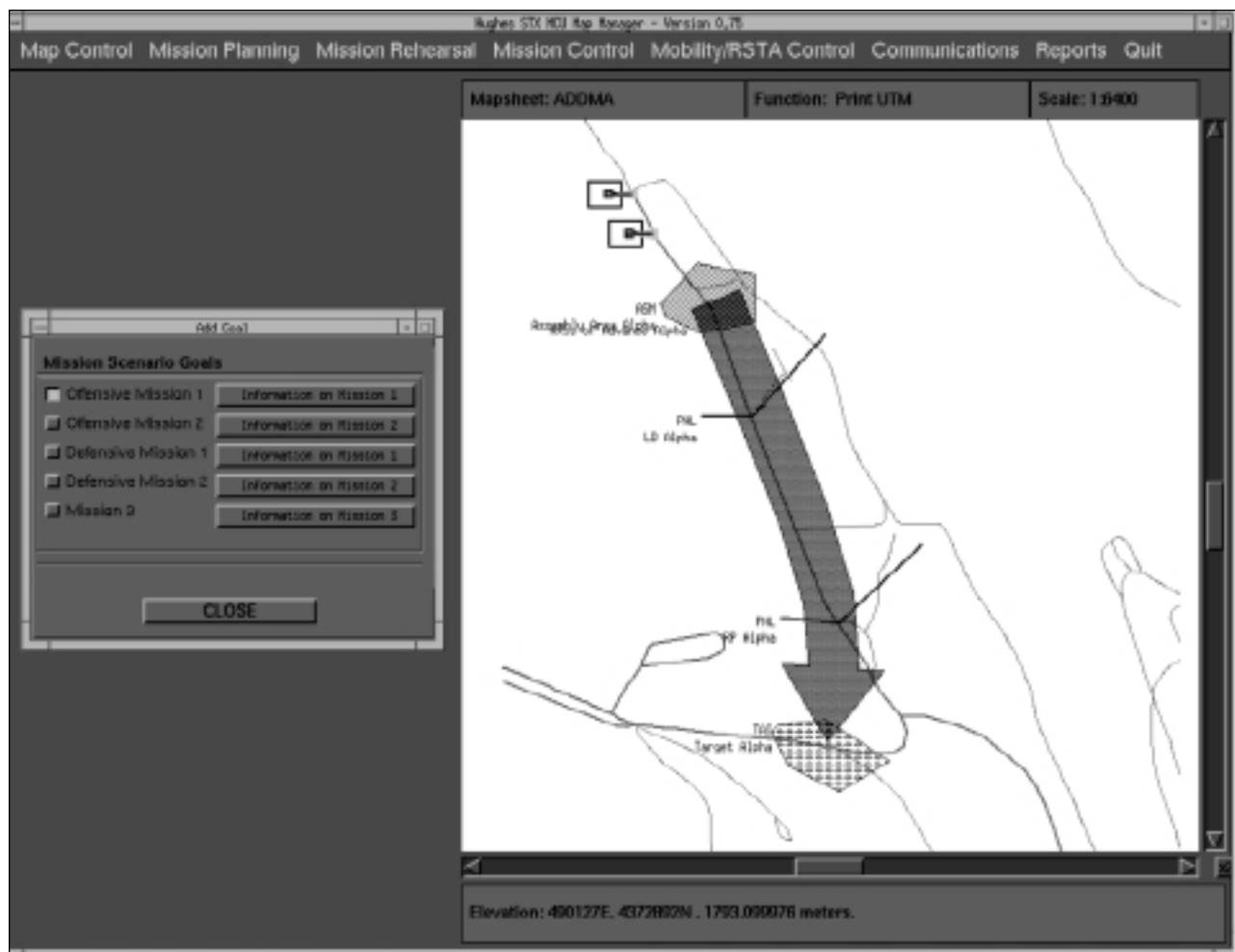


Figure 4. Mission-Planning Display.



Figure 5. Mission Execution in MODSAF.

In contrast, by incorporating knowledge into the operator's workstation and the UGVs, the demands on operator attention during mission generation and execution are greatly reduced. In cases where the mission template is already known within the workstation, it is only a matter of minutes. In addition, during execution, more responsiveness is embedded in the automation tools, allowing much more rapid (and less operator-intensive) retasking. Of course, this system had the advantage of working on a simulator, so conclusive comparisons cannot be drawn, but the approach appears promising.

It therefore appears that continual planning, such as UMPRS, can provide a powerful, tailorable substrate for operator interaction. Systems such as PRS and soar have already demonstrated the ability to control systems in problems including malfunction diagnosis and handling (Ingrand, Georgeff, and Rao 1992), aircraft sequencing (Rao and Georgeff 1995) and air combat (Tambe et al. 1995; Rao et al.

1992). At the level of multivehicle control, the approach is similar to these others; we have also demonstrated, however, that the DCP capabilities can concurrently be utilized for operator interaction to provide mixed-initiative definitions of plans and constraints. These capabilities have found other uses beyond UGVs (Durfee et al. 1997).

A related lesson is the importance of acquiring and propagating intent. Because vehicles might only intermittently be in touch with each other, it is critical that each know enough about the collective goals to forge ahead alone. The robust accomplishment of missions despite communication failures is in no small way because of the strategy of propagating goals and constraints, rather than commands, through the hierarchy. Moreover, by adopting both preplanned and run-time distributed planning techniques, agents can achieve the right blend of predictability and flexibility in how they respond to the evolving situation (Clement and Durfee 1999; Lee 1996).

Acknowledgments

This is a thoroughly revised version of an article that appeared in *Autonomous Robots* (Durfee, Kenny, and Kluge 1998), which focused more on the issues of autonomy in the man-machine system than on the DCP aspects. Nonetheless, this article borrows from that, and I am indebted to my coauthors in that work, Patrick Kenny and Karl Kluge. I would also like to acknowledge the contributions to this effort made by numerous people who have been part of our UGV team over the years, especially Marc Huber and Jaeho Lee. This work was supported, in part, by the Defense Advanced Research Projects Agency, contract DAAE07-92-CR012.

References

- Balch, T. R., and Arkin, R. C. 1998. Behavior-Based Formation Control for Multiagent Robot Teams. *IEEE Transactions on Robotics and Automation* 14(6): 926–939.
- Calder, R. B.; Smith, J. E.; Courtemanche, A. J.; Mar, J. M. F.; and Ceranowicz, A. Z. 1993. MODSAF Behavior Simulation and Control. Paper presented at the Third Conference on Computer-Generated Forces and Behavioral Representation, July, Orlando, Florida.
- Clement, B. J., and Durfee, E. H. 1999. Top-Down Search for Coordinating the Hierarchical Plans of Multiple Agents. In *Proceedings of the Third Conference on Autonomous Agents*, 252–259. New York: Association of Computing Machinery.
- Cook, D.; Gmytrasiewicz, P. J.; and Holder, L. 1996. Decision-Theoretic Cooperative Sensor Planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(10): 1013–1023.
- Durfee, E. H.; Kenny, P. G.; and Kluge, K. C. 1998. Integrated Permission Planning and Execution for Unmanned Ground Vehicles. *Autonomous Robots* 5:97–110.
- Durfee, E. H.; Huber, M. J.; Kurnow, M.; and Lee, J. 1997. taipa: Tactical Assistants for Interaction Planning and Execution. In *Proceedings of the First Conference on Autonomous Agents*, 443–450. New York: Association of Computing Machinery.
- Georgeff, M. P., and Lansky, A. L. 1986. Procedural Knowledge. *Proceedings of the IEEE* 74(10): 1383–1398.
- Huber, M. J.; Durfee, E. H.; and Wellman, M. P. 1994. The Automated Mapping of Plans for Plan Recognition. In *Proceedings of the 1994 Conference on Uncertainty in Artificial Intelligence*, 344–351. San Francisco, Calif.: Morgan Kaufmann.
- Ingrand, F. F.; Georgeff, M. P.; and Rao, A. S. 1992. An Architecture for Real-Time Reasoning and System Control. *IEEE Expert* 7(6): 34–44.
- Kluge, K.; Weymouth, T.; and Smith, R. 1994. Information Assimilation Research at the University of Michigan for the ARPA Unmanned Ground Vehicle Project. In *Proceedings of SPIE Sensor Fusion VII*. Bellingham, Wash.: International Society for Optical Engineering.
- Lee, J. 1996. An Explicit Semantics for Coordinated Multiagent Plan Execution. Ph.D. dissertation, Department of Computer Science and Engineering, University of Michigan.
- Lee, J.; Huber, M. J.; Durfee, E. H.; and Kenny, P. G. 1994. um-prs: An Implementation of the Procedural Reasoning System for Multirobot Applications. Paper presented at the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space, March, Houston, Texas.
- Rao, A. S., and Georgeff, M. P. 1995. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multiagent Systems*, 312–319. Menlo Park, Calif.: AAAI Press.
- Rao, A. S.; Moreley, M.; Selvestrel, M.; and Murray, G. 1992. Representation, Selection, and Execution of Team Tactics in Air-Combat Modeling. In *Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence*, 185–190. Hobart, Australia: Australian Computer Society.
- Shoham, Y., and Tennenholtz, M. 1994. On Social Laws for Artificial Agent Societies: Offline Design. *Artificial Intelligence* 72(1–2): 231–252.
- Stentz, A. 1995. Optimal and Efficient Path Planning for Unknown and Dynamic Environments. *International Journal of Robotics and Automation* 10(3): 89–100.
- Stentz, A. 1990. The CODGER System for Mobile Robot Navigation. In *Vision and Navigation: The Carnegie Mellon Navlab*, ed. C. E. Thorpe. New York: Kluwer Academic.
- Tambe, M.; Johnson, W. L.; Jones, R. M.; Koss, F.; Laird, J. E.; Rosenbloom, P. S.; and Schwamb, K. 1995. Intelligent Agents for Interactive Simulation Environments. *AI Magazine* 16(1): 15–39.
- Veloso, M., and Stone, P. 1998. Individual and Collaborative Behaviors in a Team of Homogeneous Robotic Soccer Agents. In *Proceedings of the Third International Conference on Multiagent Systems*, 309–316. Washington, D.C.: IEEE Computer Society Press.



Edmund H. Durfee is an associate professor of electrical engineering and computer science at the University of Michigan, where he also directs the Artificial Intelligence Lab and holds a joint appointment in the School of Information.

His research interests are in distributed AI, multiagent systems, planning, and real-time problem solving, applied to problems ranging from digital libraries to cooperative robotics, from assistance technologies to electronic commerce. He is serving as the conference chair for the Fourth International Conference on Multiagent Systems in July 2000 and is an associate editor for several journals and monograph series. His e-mail address is durfee@umich.edu.



AAAI-2000 Intelligent Systems Demonstrations

in conjunction with the Seventeenth National Conference on Artificial Intelligence

Austin Texas ■ July 30-August 3, 2000

Sponsored by the American Association for Artificial Intelligence

As the field of artificial intelligence matures, our ability to construct intelligent artifacts increases, as does the need for implemented systems to experimentally validate AI research. In addition, it is becoming more important to make the tangible results of our research accessible to each other, to the scientific community, and to the public at large. The AAAI-2000 Intelligent Systems Demonstrations program showcases state-of-the-art AI implementations and provides AI researchers with an opportunity to show their research in action.

Researchers from all areas of AI are encouraged to submit proposals to demonstrate their systems. Submissions will be evaluated on the basis of their innovation, relevance, scientific contribution, presentation, and "user friendliness," as well as potential logistical constraints. This program is primarily to encourage the early exhibition of research prototypes, but interesting mature systems and commercial products are also eligible (commercial sales and marketing activities are not appropriate in the Demonstration program, and should be arranged as part of the AAAI Exhibit Program). Demonstrations that can be used by the audience and/or that interact with the audience are particularly encouraged.

Demonstrations will be expected to be available several times during the conference. There will be several "open house" events where all demonstrations will be available, and demonstrators are urged to make their demos available at other times as much as possible. As well, each demonstration will have a scheduled and advertised time during which it is the "featured" demonstration. Each accepted demonstration system must be attended by at least one knowledgeable representative (preferably an architect of the sys-

tem) who will be available to answer in-depth technical questions at scheduled times.

Demonstration proposals must be made electronically using the forms at www.cs.rochester.edu/research/aaai2000/isd/

Researchers who cannot access the World-Wide Web may contact the organizing committee to make alternative arrangements. In addition to contact information, proposals must include the following, all of which may be submitted via the web:

- A two-page description in AAAI paper format of the technical content of the demo, including credits and references.
- A 150-word summary of the demo in plain text. Please include title, demonstrators, and affiliation. This summary will be used to compile a program for the Demonstrations.
- An informal videotape of the demo (in NTSC VHS format), or a demo storyboard of not more than six pages total. This is the committee's primary method of evaluating your proposal. Videotapes (three copies) should be mailed to the address given on the web page.
- A detailed description of hardware and software requirements. Demonstrators are encouraged to be flexible in their requirements (possibly with different demos for different logistical situations). Please state what you can bring yourself and what you absolutely must have provided. Generally speaking, we can provide generic PCs with standard software such as web browsers, computer monitors, and peripherals such as TVs and VCRs. We will do our best to provide resources but nothing can be guaranteed at this point beyond space and power.
- For demonstrations accessible via the web, we hope to maintain a page of links so that users can try the systems before or after the conference. Anyone interested in participating should include a URL that accesses their demo with their proposal.

Demo proposals must be received in their entirety including any supporting materials by Friday, February 25, 2000. Authors will be notified of acceptance by April 3, 2000.

We especially hope that authors of papers accepted for presentation at the conference technical program will be able to demonstrate their research in the Intelligent Systems Demonstration Program. To present a system demonstration, however, the authors must still submit a proposal conforming to the above requirements by the Demonstration program deadline.

Submitters who wish to demonstrate intelligent mechanical systems that interact with the real world (aka "robots") should direct their efforts toward the AAAI Robot Exhibition.

If you have any questions or comments about the Intelligent Systems Demonstration program, we encourage you to address them to the program committee:

- George Ferguson
(ferguson@cs.rochester.edu)
- Chris Welty (welty@ladseb.pd.cnr.it)
- K. Suzanne Barber
(barber@mail.utexas.edu)