# AI planning systems in the real world

Planning systems generate partially ordered sequences of actions (or plans) that solve a goal. They start from a specification of the valid actions (also called operators), which includes both the conditions under which an action applies (the preconditions) and the expected outcome of applying that action (the effects). The general problem is quite hard both because of the potentially enormous search space and the difficulty in fully and accurately representing real-world problems. Approaches to planning include operator-based planning, hierarchical task-network planning, case-based planning, reactive planning, and many more. Early planning work focused largely on "toy" problems (for example, the blocks world). More recently, there has been a big push toward applying planning systems to real-world applications. While planning systems have not yet achieved the level of commercial success enjoyed by some other areas of artificial intelligence—neural nets, for example—a number of successful applications of planning technology to real-world problems have recently emerged.

This installment of "Trends & Controversies" highlights five such applications. I have asked the developers of these systems to describe the application domain and the planning technology used to solve the problems. These systems all use some form of hierarchical task-network planning (in some cases combined with other techniques). HTN planning provides a way of specifying, as part of the operator definition, how to hierarchically expand actions into partially ordered sequences (task networks) of actions. This approach succeeds, in part, because it provides a natural way of limiting the possibly very large search spaces. See *Readings in Planning* (Morgan Kaufmann, 1990) or *Artificial Intelligence: A Modern Approach* (Prentice Hall, 1995) for more details on various planning techniques.

In the first article, Stephen Smith, Dana Nau, and Thomas Throop describe their use of planning technology to build a system for declarer play in contract bridge. The system can beat the best commercially available program and is currently being incorporated into a commercial product. Second, John Mark Agosta and David Wilkins describe how the SIPE-2 planner helps evaluate the US Coast Guard's ability to respond to marine oil spills. This system, which automates a problem that is currently done by hand, is undergoing evaluation by the Coast Guard. Third, Austin Tate describes a planning application, in use by the European Space Agency, for the project management of spacecraft assembly, integration, and verification. Fourth, Steve Chien and his colleagues describe their use of a planning system to automate the operations of NASA's Deep Space Network communication antennas. This system is currently being integrated into a new system that will become operational in 1997. Finally, Thomas Lee and David Wilkins describe their use of SIPE-2 in producing military air campaign plans. Their planner is part of a demonstration system that is fully integrated with the other software modules currently used for solving parts of this problem.

***—Craig Knoblock***

## AI planning's strong suit

*Stephen J.J. Smith, Hood College*
*Dana Nau, University of Maryland*
*Thomas Throop, Great Game Products*

Although game-tree search techniques work well in perfect-information games such as chess, checkers, and Othello, difficulties arise in adapting them to imperfect-information games such as bridge. In bridge, the game tree's branching factor is very large because no player has complete knowledge about the state of the world, the possible actions, and their effects. Because bridge deals must be played in just a few minutes, a full game-tree search will not search a significant portion of this tree within the time available. Matthew Ginsberg is developing a modified game-tree search procedure to address this problem.[1] However, others have shown some pitfalls in any approach that (like Ginsberg's) treats an incomplete-information problem as a collection of complete-information problems.[2] No evidence yet proves that these pitfalls can be overcome.

Our approach grows out of the observation that bridge is a game of planning. The bridge literature describes a number of tactical and strategic schemes (such as finessing, ruffing, and crossruffing) that people combine into plans in playing bridge games. We have taken advantage of the planning nature of bridge, by adapting and extending some ideas from HTN planning.

### Approach

HTN planning is an AI planning methodology that creates plans by *task decomposition*—by decomposing tasks into smaller and smaller subtasks until primitive tasks are found that can be performed directly. HTN planning systems have knowledge bases containing *methods* that tell how to develop plans by such decompositions.[3–5] Given a task to accomplish, the planner chooses an applicable method and instantiates it to decompose the task into subtasks, and chooses and instantiates other methods to decompose the subtasks even further. If the constraints on the subtasks or the interactions among them prevent the plan from being feasible, the planning system will backtrack and try other methods.

To represent the tactical and strategic schemes of card playing in bridge, we use structures similar to the methods described just now, but modified to represent multiagency and uncertainty. For example, Figure 1 shows a portion of our task-network structure for a bridge tactic called *finessing.* To generate game trees, we use a procedure similar to task decomposition to build up a game tree whose branches represent moves generated by these methods.

For a game tree generated in this manner, the number of branches from each state is *not* the number of actions an agent can perform (as in conventional game-tree search procedures), but instead is the number of different tactical and strategic schemes the agent can employ. This results in a smaller branching factor and a much smaller search tree: Tignum 2 generates game trees containing only about 420,000 nodes in the worst case and 26,000 nodes on the average, as compared to $6 \times 10^{44}$ nodes in the worst case and $10^{29}$ nodes on the average if we had generated a conventional game tree. Thus, Tignum 2 can search the game tree all the way to the end, to predict the likely results of the various sequences of cards it might play.[6,7]

### Comparison with conventional HTN planning

In Tignum 2, we have extended HTN planning to include ways to represent and reason about possible actions by other agents (such as the opponents in a bridge game), as well as uncertainty about their capabilities (for example, lack of knowledge about what cards they have). However, to accomplish this, we needed to restrict how Tignum 2 formulates its plans. Most HTN planners develop plans in which the actions are only partially ordered, postponing some of the decisions about the order in which the actions will be performed. In contrast, Tignum 2 is a total-order planner that expands tasks in left-to-right order.

Because Tignum 2 expands tasks in the same order in which they will be performed when the plan executes, this means that when it plans for each task, Tignum 2 already knows the state of the world (or as much as can be known about it in an imperfect-information game) at the time that the task will be performed. Consequently, we can write each method's preconditions as arbitrary computer code, rather than using the stylized logical expressions found in most AI planning systems. This enables us to encode the complex numeric computations needed for reasoning about the probable locations of the opponents' cards. For example, by knowing the current state, Tignum 2 can decide which of 19 finesse situations are applicable: with partial-order planning, it would be much harder to decide which of them *can be made* applicable.

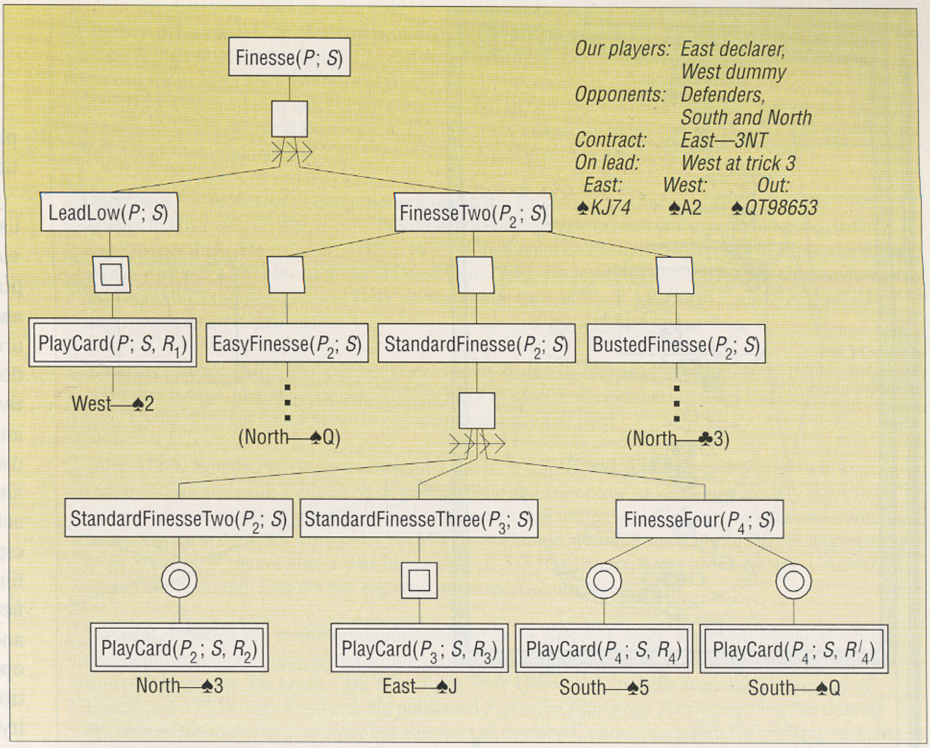



Figure 1. A portion of a finesse method.

## Performance

To test Tignum 2, we played it against Bridge Baron, from Great Game Products. Winner of a number of important bridge competitions, Bridge Baron is probably the best program for declarer play at contract bridge. In reviewing seven commercially available bridge-playing programs, the American Contract Bridge League rated Bridge Baron best and also best of the five that do declarer play without "peeking" at the opponents' cards.[8]

When we tested Tignum 2 against Bridge Baron on 1,000 randomly generated bridge deals (including both suit and no-trump contracts), Tignum 2 beat Bridge Baron by 254 to 202, with 544 ties. These results are statistically significant at the $\alpha = 0.05$ level. We had never run Tignum 2 on any of these deals before this test, so these results are free from any training-set biases.

## Conclusions

The use of HTN planning techniques in Tignum 2 enables it to do bridge declarer play better than Bridge Baron. Tignum 2 is being incorporated into Bridge Baron to improve the Baron's declarer play.

We have been quite successful in using the same modified version of HTN planning (as well as some of the same code!) in another very different application domain: the task of generating process plans for the manufacture of complex electro-mechanical devices.[9] That this same approach works well in two such widely varying areas is quite striking, suggesting that our approach may be useful in a number of practical planning problems.

## Acknowledgments

This work was supported in part by an AT&T PhD scholarship to Stephen J. J. Smith, by Maryland Industrial Partnerships (MIPS) Grant 501.15, by Great Game Products, by ARPA Grant DABT 63-95-C-0037, and by National Science Foundation Grants NSF EEC 94-02384 and IRI-9306580.

## References

1. M. Ginsberg, "How Computers Will Play Bridge," to appear in *Bridge World*.
2. I. Frank and D. Basin, "Search in Games with Incomplete Information: A Case Study Using Bridge Card Play," Research Paper 780, Dept. of AI, Univ. of Edinburgh, Edinburgh, Scotland, 1995.
3. D.E. Wilkins, *Practical Planning*, Morgan Kaufmann, San Francisco, 1988.
4. K. Erol, J. Hendler, and D.S. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning," *Proc. Second Int'l Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1994, pp. 249–254.
5. E.D. Sacerdoti, *A Structure for Plans and Behavior*, American Elsevier, New York, 1977.
6. S.J.J. Smith, D.S. Nau, and T. Throop, "A Planning Approach to Declarer Play in Contract Bridge," *Computational Intelligence*, Vol. 12, No. 1, Feb. 1996, pp. 106–130.
7. S.J.J. Smith, D.S. Nau, and T. Throop, "Total-Order Multiagent Task-Network Planning for Contract Bridge," *AAAI-96*, AAAI Press, 1996, pp. 108–113.
8. B. Manley, "Software 'Judges' Rate Bridge-Playing Products," *The Bulletin*, Vol. 59, No. 11, Nov. 1993, pp. 51–54.
9. K. Hebbar et al., "Plan-Based Evaluation of Designs for Microwave Modules," *ASME. Design for Manufacturing Conf.*, Am. Soc. Mechanical Engineers, New York, 1996, p. 262.

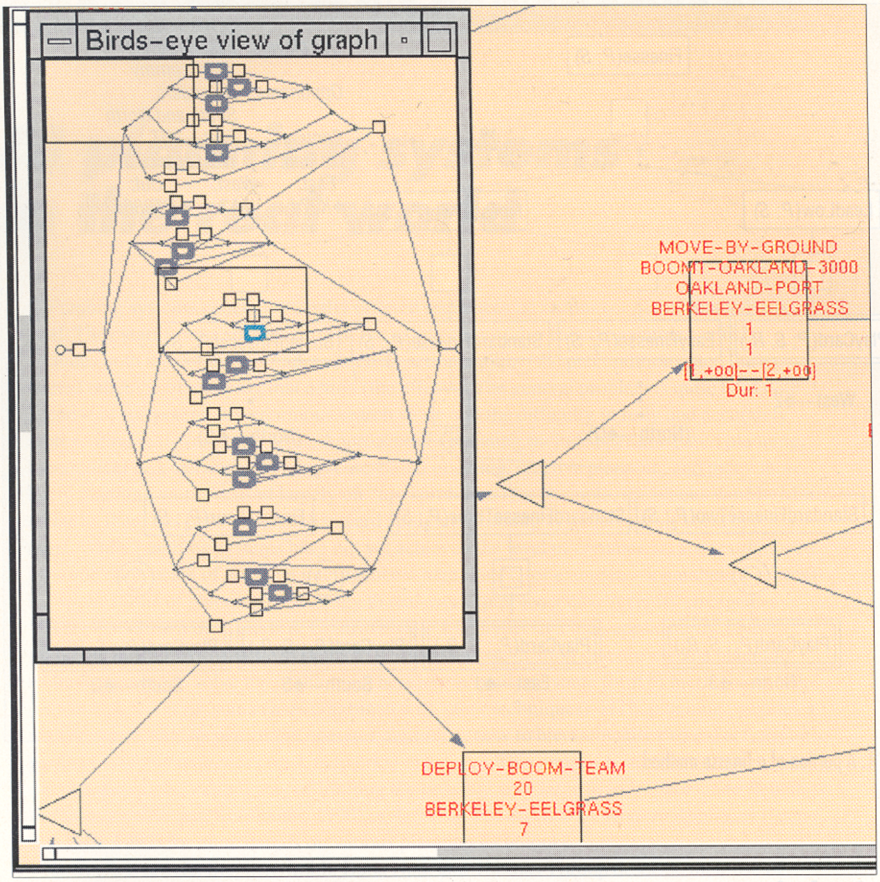



Figure 2. SIPE-2 displays the ordering of actions in the completed plan. The bird's-eye view shows the entire plan. The enlarged portion of the plan on the screen shows actions to accumulate and deploy boom.

## Using SIPE-2 to plan emergency response to marine oil spills

*John Mark Agosta and David E. Wilkins, SRI International*

The objective of the Spill Response Configuration System built by SRI International is to help the US Coast Guard estimate the adequacy of the amounts and locations of cleanup equipment in its coastal oil-spill incident response plans. SRCS guides the user in building plans that meet a range of spill scenarios and then evaluates the plans. This research project is a novel application of automated planning. Previous approaches used rule-of-thumb planning factors to estimate equipment needs, such as quantity of containment booms, based on the size and frequency of spills. By automating some of the planning steps, SRCS lets users plan and evaluate a range of detailed responses to a range of spill scenarios, enabling the USCG to more accurately estimate its needs.

SRCS integrates simulation, evaluation, map display, and scheduling tools with the System for Interactive Planning and Execution (SIPE-2) planner. For SRCS, we built a knowledge base so that SIPE-2 can generate oil-spill response plans interactively. SRCS is intended to be used for configuration planning—planning done in preparation of likely incidents—rather than planning done as an incident unfolds. The system leveraged technology from previous SRI work on military course-of-action planning under the DARPA-funded System for Operations Crisis Action Planning (Socap) project.[1]

### The planning technology

SIPE-2 is an AI planning system that supports planning at multiple levels of abstraction to generate partially ordered plans. It provides a formalism for describing actions as operators and utilizes knowledge encoded in this formalism, together with heuristics for reducing the computational complexity of the problem, to generate plans for achieving given goals. Given an arbitrary initial situation, the system either automatically or under interactive control combines operators to generate plans that achieve the prescribed goals, thus producing a novel sequence of actions that responds precisely to the situation at hand. The generated plans include information that allows the system to respond to unexpected occurrences during plan execution and to modify its plans to take these into account.

SIPE-2 provides a powerful graphical user interface to aid in generating plans, viewing complex plans and other information as graphs on the screen, and following and controlling the planning process. Figure 2 shows the part of the plan network for deploying containment boom in one sensitive area. The SIPE-2 technology is generic and domain-independent; it has proven useful on a large variety of problems. Example applications include planning the actions of a mobile robot, managing aircraft on a carrier deck, air campaign planning, construction tasks, producing products from raw materials under production and resource constraints, and joint military operations planning. Two of these applications have been used in integrated feasibility demonstrations of the DARPA-Rome Lab Planning Initiative.

### Handling numeric goals

To support this application, we made one major enhancement to SIPE-2, which enables it to achieve metric goals for accumulating a certain level of something. An example is a goal to accumulate several thousand feet of oil-containment boom to protect a sensitive area. This goal must typically be met by transporting several shipments of boom from different locations.

SIPE-2 could already reason about resource production and consumption. The challenge of metric accumulation goals is that they are not accomplished by a single action; rather, several actions contribute to the accumulation. Therefore, we extended SIPE-2 to solve a goal by adding a set of ordering links (previously, it could add only a single ordering link). For example, 11 parallel actions might together solve an accumulation goal. The computation of finding a suitable set of actions is inefficient in the worst case, but has been fast in practice. This capability was proven useful in other applications as well.

### Design of the system

SRCS's overall design comprises five main modules: equipment deployment planner, plan scheduler, trajectory and oil disposition model, evaluation module, and color map display.

By its nature, oil-spill incident response is a race against time, to contain or remove oil before it damages the shore. Planning begins by entering the specifics of a spill incident—location, time of day, spill rate, and so on—then forecasting the spill trajectory, considering the uncertainty in its spreading caused by wind and waves. This forecast determines which environmentally sensitive shore sectors the oil will hit, and when. The planner works from this forecast, together with geographic information, such as the sectors into which the region is divided and the USCG requirements for protection of these areas. In addition, the planner works with the database of the quantities and capabilities of available equipment and resources, and where they are located. The planner, SIPE-2, and scheduler, Tachyon, then work interactively with the user to generate a plan of equipment deployment and employment actions that meet constraints among oil spreading, equipment cleanup capabilities and transport times, and environmental protection requirements. Finally, the evaluation module uses the scheduler output and the projected flows from the trajectory model to determine the effectiveness of the plan.[2]

Most of the user's interaction with SRCS is mediated by a map interface, implemented in the Arcview commercial geographical information system. The user thus can immediately see both the extent of the spill and where resources are employed at various times.

## Evaluating the plan

In the SRCS domain, plans are distinguished by the degree to which they achieve the overall objective of cleaning up the spilled oil. In many spills, much of the oil will escape, no matter how much equipment is available, because of the difficulty of operations and speed of spreading due to the weather. Furthermore, for any spill, SRCS can generate many possible plans, and users can partially or completely sacrifice a sector cleanup goal if they believe equipment that would have been assigned to a sector better serves the overall goals by being used elsewhere. The plan and the oil flows determined by the trajectory model become the input to the evaluation model. The evaluation model accounts for the quantities of oil contained and removed in each sector, for each period. From this accounting, it can calculate measures of plan merit, such as the final fraction of oil removed under each plan.

Because the evaluation model is graphic and efficiently computed, SRCS has


**Stephen J.J. Smith** is an assistant professor at Hood College. His research interests include task-network planning, forward pruning, and search. He earned a BS in computer science, mathematics, and Latin from Dickinson College and an MS in computer science from the University of Maryland, College Park, and is expecting his PhD in computer science from the University of Maryland. Contact him at the Dept. of Mathematics and Computer Science, Hood College, 401 Rosemont Ave., Frederick, MD 21701-8575; sjsmith@nimue.hood.edu.

**Dana Nau** is a professor of computer science at the University of Maryland's Institute for Systems Research. His research interests include computer-integrated manufacturing, AI planning, and search algorithms. He received a BS in applied mathematics from the University of Missouri-Rolla and an AM and a PhD in computer science from Duke. He is a fellow of the AAAI. Contact him at the Dept. of Computer Science, Univ. of Maryland, College Park, MD 20742; nau@cs.umd.edu; http://www.cs.umd.edu/~nau.

**Thomas Throop** is the president of Great Game Products, maker of Bridge Baron, the leading computer bridge-playing program. He has a BS in electrical engineering from Swarthmore College. Contact him at (301) 365-5277; greatgames@delphi.com.

**John Mark Agosta** is a computer scientist in SRI's Applied AI Technology Program. He specializes in the formulation and design of probabilistic and economic models, specifically Bayesian and decision-theoretic methods applied to automated planning and decision making under uncertainty. He earned his BS from Yale, his MA from George Washington, and his PhD in engineering from Stanford. Contact him at SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025; johnmark@sri.com; http://www.erg.sri.com/people/johnmark/.

**David E. Wilkins** is a senior computer scientist at the SRI AI Center, where his research focuses on planning and reasoning about actions, knowledge representation, and design and implementation of AI systems. He holds a BS from Iowa State University, an MSc from the University of Essex, and a PhD from Stanford. He published *Practical Planning: Extending the Classical AI Planning Paradigm*, and recently led a project to develop Cypress, a system for creating taskable, reactive agents. Contact him at SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025; wilkins@ai.sri.com; http://www.ai.sri.com/~wilkins.

**Austin Tate** is technical director of the Artificial Intelligence Applications Institute and holds the Personal Chair of Knowledge-Based Systems at the University of Edinburgh. Besides engaging in the research, development, and application of knowledge-based methods, he also has a background in databases and software engineering. He holds a PhD in machine intelligence from the University of Edinburgh. He is an editorial board member of *IEEE Expert*. Contact him at the AIAI, Univ. of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; a.tate@ed.ac.uk; http://www.ed.ac.uk.

**Steve Chien** is technical group supervisor of the AI Group of the Jet Propulsion Laboratory, California Institute of Technology, where he leads efforts in research and development of automated planning and scheduling systems. He is also an adjunct assistant professor in the Department of Computer Science at the University of Southern California. He holds a BS, MS, and PhD in computer science from the University of Illinois. His research interests are in the areas of planning and scheduling, operations research, and machine learning. Contact him at steve.chien@jpl.nasa.gov; http://www-aig.jpl.nasa.gov.

**Anita Govindjee** is a member of the technical staff in JPL's AI Group. She holds an MS in computer science from Stanford and a BS in computer science from the University of Illinois. Her research interests are in AI and cognitive science. Contact her at anita.govindjee@jpl.nasa.gov.

**XueMei Wang** is a member of the technical staff at the Rockwell Science Center in Palo Alto, California. She holds a PhD in computer science from Carnegie Mellon and a BS in applied mathematics from Tsinghua University. Her research interests are AI—including planning, machine learning, and knowledge acquisition. Reach her at mei@rpal.rockwell.com.

**Tara Estlin** is a PhD candidate in computer science at the University of Texas at Austin. She holds a BS in computer science from Tulane University and an MS in computer science from the University of Texas at Austin. Her research interests include machine learning and planning and scheduling. Contact her at estlin@cs.utexas.edu.

**Randall Hill Jr.** is a research computer scientist at the Information Sciences Institute, University of Southern California, and also a research assistant professor in the USC Department of Computer Science. He holds a BS from the United States Military Academy at West Point, and MS and PhD degrees in computer science from USC. His research interests lie in the areas of intelligent agents and computer-assisted learning environments. Reach him at hill@isi.edu.

**Thomas J. Lee** is a senior research engineer at SRI International. His research interests include crisis planning, machine learning, and game playing. He received an MS in computer science from the University of Wisconsin and a BS in computer science from the University of Nebraska. He can be reached at SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, tomlee@erg.sri.com.

attracted interest as a research tool. It has been used in several experiments, such as for machine learning of operator preconditions, for decision-theoretic planning, and to optimize over a range of partial plans generated by SIPE-2, based on which plan removes oil better.

This optimization was conducted by manually building an influence diagram model for a partially constrained plan, then solving the model to determine what distribution of equipment maximized the overall fraction of oil cleaned up.[3]

## The future of SRCS

We have periodically demonstrated SRCS to the USCG during its development, as we added features and modules. We demonstrated it at USCG port exercises and in conjunction with the manual configuration response-planning exercises that USCG-run area committees are required to conduct. SRCS is ready for limited fielding, to tailor its interface design and validate its knowledge base.

SRSC could be extended to assist during an incident. The incident command's planning team could develop comprehensive plans that are adjusted, as needed, as the situation evolves, and conveyed to the operations teams. Many experts who have seen SRCS recognize the value of eventually extending it to a real-time response planning system.


### Acknowledgments

This work was supported under US DOT Grants DTRS-57-94-Y-00081 and DTRS-57-94-G-00084.


### References


1. M. Bienkowski, "Oil-Spill Contingency Evaluation," in "Demonstrating the Operational Feasibility of New Technologies," *IEEE Expert*, Vol. 10, No. 1, Feb. 1995, p. 30.

2. J.M. Agosta, "Formulation and Implementation of an Equipment Configuration Problem with the SIPE-2 Generative Planner," *Proc. AAAI-95 Spring Symp. Integrated Planning Applications*, AAAI Press, Menlo Park, Calif., 1995, pp. 1–10.

3. J.M. Agosta, "Constraining Influence Diagram Structure by Generative Planning: An Application to the Optimization of Oil Spill Response," *Proc. 12th Conf. Uncertainty in AI*, AAAI Press, 1996, pp. 11–19.


## Responsive planning and scheduling using AI planning techniques—Optimum-AIV

*Austin Tate, AIAI, University of Edinburgh*

Simple AI planning techniques involving the refinement of higher-level actions in a plan into lower-level expansions have found their way into many computer systems. However, there is some question over whether more elaborate techniques developed by the AI planning community have achieved the sort of widespread application now being seen, for example, for AI scheduling techniques.

Experience at the Artificial Intelligence Applications Institute at the University of Edinburgh indicates that there are applications of a range of powerful AI planning techniques, but that these are not as yet as widespread as those for more narrowly focused scheduling and constraint-management techniques.

This essay documents one example of a deployed planning aid based on a number of techniques that have been developed by the AI planning community. The primary example described is the Optimum-AIV system, which was developed to assist with the project management of the assembly, integration, and verification of spacecraft such as ERS-1 at the European Space Agency.

As background, AIAI had worked with European partners CRI (Denmark) and Matra Espace (France) on earlier planning systems for ESA, such as PlanERS, a mission planer for ERS-1 (see Figure 3). With the addition of ProgesSpace (France), the same team was asked to build a deployable system as a result of these early demonstrations. AIAI was responsible for the plan representation used and object-oriented designs for the primary planning and test-failure recovery planning algorithms. CRI acted as systems integrator and implementor for the whole system. AIAI drew on earlier work with Nonlin[1] and O-Plan,[2] and on experience in using knowledge-rich plan representations to augment commercial process planners, project managers, and job-shop schedulers on the Planit project, an effort involving 26 organizations in the UK.[3]

### Optimum-AIV

Planning is a key issue in the management of a space project's assembly, integration, and verification (AIV) activities. Not only must technological requirements be met, but cost and time are critical. There are costly testing facilities, which must be shared with other projects, and planning needs to occur to coordinate between a number of participants (agencies, contractors, launcher authorities, and users). A delay caused by one participant normally leads to serious problems for others. Space project managers at all levels are concerned with planning, and they closely control the work's progress. However, finding computer-based planning aids that meet the needs of this application has been difficult. General-purpose project management software cannot represent the wide range of factors to be taken into account, and are too complex for interactively modifying plans during project execution.[4] Thus, the ESA commissioned the Optimum-AIV system, which utilizes AI planning representations and techniques.

Optimum-AIV, as a deployed system, was concerned with the integration of AI planning methods into an existing project management environment based on the use of the commercial Artemis project management tool (the developer of Artemis was a member of the Planit project with AIAI). Much of the project concerned user-interface and integration issues. However, the plan representation used and the algorithms and aids that could be added because of the rich plan representation were an important advance. The applied AI planning techniques adopted complemented in a natural way those facilities already available via Artemis. Details of Optimum-AIV and the techniques are available in *Intelligent Scheduling*,[5] from which extracts appear in the list of methods below.

- Optimum-AIV adopts a partially ordered plan representation, which supports causally independent activities that can execute concurrently.
- It searches through a space of partial plans, modifying them until it finds a valid plan or schedule.

## Coming in February's Trends & Controversies

**Indexing the World Wide Web**

- Michael Mauldin, Lycos Inc.
- Oren Etzioni and Erik Selberg, University of Washington

- The system employs hierarchical planning. *Hierarchical* refers to both the representation of the plan at different levels, and also the control of the planning process at progressively more detailed levels.
- During plan specification and generation, the system operates on explicit preconditions and effects of activities that specify the applicability and purpose of the activity within the plans. With this knowledge, it is possible to check whether the plan's current structure introduces any conflicts between actual spacecraft system states, computed by the system, and user-specified activity preconditions. Such conflicts would arise if one activity deleted the effect of another, thus removing its contribution to the success of a further activity. The facility for checking the plan logic's consistency, by dependency recording, is not possible within existing project management tools, which assume that the user must get this right.
- Detailed constraints are associated with the plan. These represent resource and temporal constraints on the activities in the plan, as well as a more general class of global activity constraints. The scheduling task in Optimum-AIV is considered as a constraint-satisfaction problem solved by constraint-based reasoning. The constraints propagate throughout the plan, gradually transforming it into a realizable schedule. Invariably, not all of the constraints can be met, such that some have to be relaxed via user intervention.
- During planning, the system records the rationale behind the plan structure; that is, user decisions on alternatives are registered. This assists in plan repair during which the user tries to restore consistency. The user can then derive information about alternative activities, soft constraints that may be relaxed, and potential activities that may be performed in advance.
- Test failure recovery plans are available as plan fixes for bringing the plan back on track after a test fails during the assembly and integration process. The same AI planning methods used to generate a plan also assist in fixing such problems. Optimum-AIV assists the user in plan repair in an interactive way rather than by performing the repair itself.

Optimum-AIV is in use for planning the production of the European Ariane-4 launcher's vehicle equipment bays.[4] The Araine-4 project team chose the system because of

- the wealth of information that can be provided to and used by the tool to describe the constraints inherent in the AIV activity;
- the quality of support provided by the tool to allow resource conflicts to be resolved;
- the clear representation of information and of the interactive capabilities, which enables engineering management to access several planning scenarios on line; and
- the fact that Optimum-AIV provides a single solution to problems of managing the plan, schedule, and allocation of resources among competing vehicle equipment bays that are concurrently being assembled.

Optimum-AIV provides a rich plan representation and aids to allow for the editing of AIV planning information and a wide range of constraints on the process. This information forms a basis for plan generation, checking of plan logic, and analysis of plans. Facilities are available to allow for the interactive repair of executing plans when tests indicate failures of components under assembly and integration. Optimum-AIV is an example of a deployed application of a number of AI planning techniques.

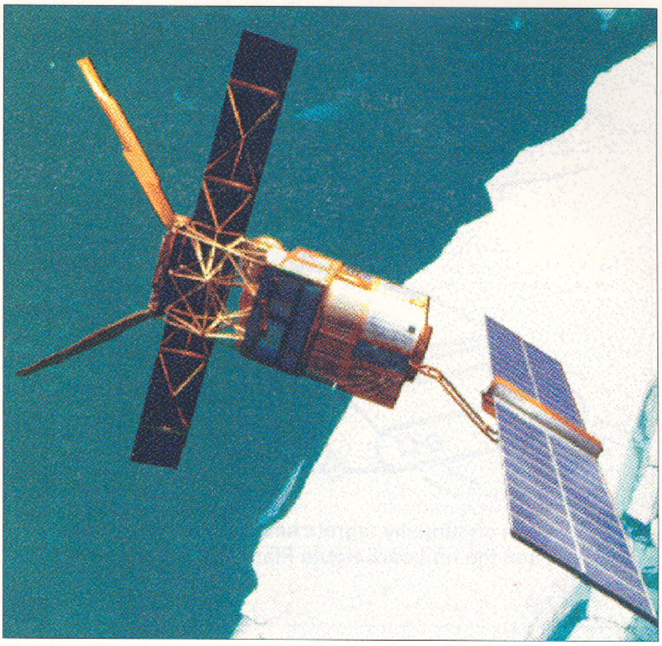


Figure 3. Artist's rendition of the ERS-1 spacecraft. (Photo courtesy of the European Space Agency.)

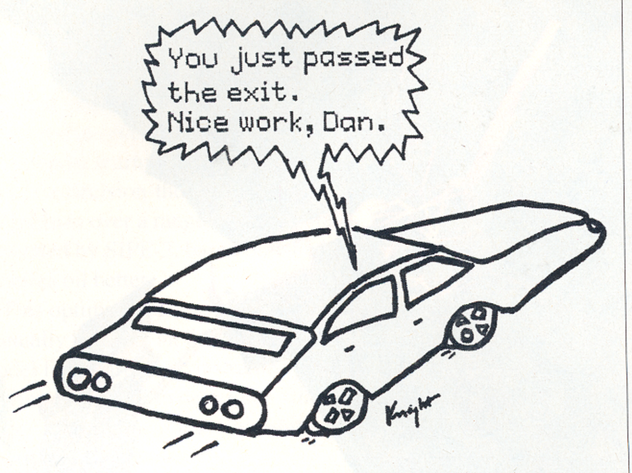
### References


1. A. Tate, "Generating Project Networks," *Proc. Int'l J. Conf. AI*, Morgan Kaufmann, San Francisco, 1977, pp. 888–893.
2. K. Currie and A. Tate, "O-Plan: The Open Planning Architecture," *Artificial Intelligence*, Vol. 52, 1991, pp. 49–86.
3. M.E. Drummond and A. Tate, "PLANIT Interactive Planners' Assistant—Rationale and Future Directions," available as AIAI-TR-108, AIAI, University of Edinburgh, Edinburgh, UK, 1992.
4. Y. Parrod and S. Valera, "Optimum-AIV, a Planning Tool for Spacecraft AIV," *Preparing for the Future*, Vol. 3, No. 3, 1993, pp. 7–9, also available through http://www.aiai.ed.ac.uk/~pasg/Projects/optimum_home.html.
5. M. Aarup et al., "Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV," *Intelligent Scheduling*, M. Zweben and MS Fox, eds., Morgan Kaufmann, 1994, pp. 451–469.


## Integrating hierarchical task-network and operator-based planning techniques to automate operations of communications antennas


*Steve Chien, Anita Govindjee, XueMei Wang, Tara Estlin, and Randall Hill Jr., Jet Propulsion Laboratory*


Since it was established in 1981, the Deep Space Network has evolved into the world's largest, most sensitive scientific telecommunications and radio navigation network. Its purpose is to support unpiloted interplanetary spacecraft missions as well as radio and radar astronomy observations in the exploration of the solar system and the universe. There are three deep-space communications complexes, located in Canberra, Australia; Madrid, Spain; and Goldstone, California. The DSN receives




**Dan continually regrets having typed his name into the on-board Route Planning Computer.**

telemetry signals from spacecraft, transmits commands that control the spacecraft operating modes, generates the radio navigation data used to locate and guide the spacecraft to its destination, and acquires flight radio science, radio and radar astronomy, very long baseline interferometry, and geodynamics measurements.

From its inception, the DSN has been driven by the need to create increasingly more sensitive telecommunications devices and better techniques for navigation. Operating the DSN communications complexes requires a high level of manual interaction with the devices involved in communication links with spacecraft. More recently, NASA has added new goals to the development of the DSN:

- reduce the cost of operating the DSN;
- improve the operability, reliability, and maintainability of the DSN; and
- prepare for a new era of space exploration with the New Millennium program by supporting small, intelligent spacecraft requiring very few mission operations personnel.

Each day, at sites around the world, NASA's DSN antennas and subsystems perform scores of tracks to support earth-orbiting and deep-space missions. Because of the equipment's complexity, the large set of communications services (in the tens), and the large number of supported equipment configurations (in the hundreds), correctly and efficiently operating this equipment to fulfill tracking goals is daunting. The antenna operations knowledge embodied in the system also must be easily understandable and maintainable as equipment, services, protocols, and software changes evolve.

The DSN Antenna Operations Planner (Dplan) is an automated planning system developed by the Jet Propulsion Laboratory to automatically generate antenna tracking plans to satisfy DSN service requests. To generate these antenna operations plans, Dplan uses a number of information sources, including the project-generated service request, the spacecraft sequence of events, the track equipment allocation, and an antenna operations knowledge base. The service request represents the basic communications services requested during the track (telemetry/downlink, commanding/uplink, ranging—uplink and downlink, and so forth). The sequence of events indicates the relevant spacecraft mode changes (such as transmission bit rate changes and modulation index changes).

The equipment allocation dictates the antenna and subsystem configuration available for the track. The antenna operations knowledge base provides information on the requirements of antenna operations actions; in particular, this information dictates how these actions can be combined to provide essential communications services. Dplan uses AI planning techniques to synthesize the operations plans. Dplan uses both HTN planning techniques and operator-based planning techniques. In HTN planning, abstract actions such as "calibrate receiver" or "configure sequential ranging assembly" decompose into specific directives for specific hardware types. In operator-based planning, requirements of specific actions are satisfied using means-end analysis, which matches action preconditions to effects and resolves any occurring ordering constraints.

By using a combination of HTN and operator-based planning techniques, Dplan can succinctly represent the complexity of the antenna operations domain. In this integration, HTN rules provide general templates for achieving goals in the context of certain equipment types. For example, to fulfill the very long baseline interferometry (VLBI) service, the required steps for receiver calibration depend on whether the track has been allocated a Block IV (older) receiver or a Block V (newer) receiver. This dependency is represented by having two HTN rules—one of which applies to each receiver type.

In addition to this variability, constraints on steps relating to other subsystems—the metric data assembly and microwave controller—depend on the receiver calibration procedure. Dplan allows the basic structure of the receiver calibration to be represented in the HTN rule.

However, many lower-level links may depend on the exact combination of goals and equipment assigned. Representing these interaction combinations completely in HTN rules would require on the order of a rule for each different combination of interactions. Dplan allows representation of these context-dependent interactions using operator-based links (preconditions and effects that are linked differently in different cases). Leaving these interactions to be resolved by the operator-based planner produces a simpler, more readable knowledge base. Using HTN rules to produce the general template makes the overall structure of produced plans clearer to the knowledge engineers by allowing "almost modular" pieces to be represented modularly in the HTN rules, thus increasing the understandability of the knowledge base. Also, using HTN rules this way reduces search during plan construction. Figure 4 shows a plan constructed by Dplan to perform precalibration of a 34-meter beam waveguide antenna for a telemetry, commanding, and ranging track.

Dplan was initially demonstrated in February 1995 for Voyager downlink, telemetry tracks at the DSS-13 antenna at Goldstone. NASA is currently integrating Dplan into the network monitor and control upgrade being deployed at DSN stations and is scheduled to become operational in February 1997. The current Dplan system supports the full range of 34-m and 70-m antenna types, all standard service request classes, and approximately 20 subsystems. Dplan covers this large class of problems with a relatively compact knowledge base (current size: 97 HTN rules and 106 activity/operator definitions).

In addition to classical planning, DSN antenna operations also require replanning, which occurs in two general cases: changing goals or changing state. First, shortly before or during a track, a project sometimes might submit a request to add services to a track. These correspond to additional goals to be incorporated into the track plan. Currently, Dplan does not automatically handle added services: it is an area of future work.

Second, after a plan has been generated, the problem state (equipment availability and status) might change. For example, a block (plan step) might fail (presumably due to equipment failure), a piece of equipment might require resetting (due to general unreliability), or a piece of equipment might be faulty or be preempted by a higher priority track. For a simple plan-step failure, Dplan simply calls for reexecution of the block. If a piece of equipment requires resetting, Dplan has knowledge describing which achieved goals are undone and require reestablishment. Dplan then uses a replanning technique that reuses parts of the original plan as necessary to reachieve the undone goals. This technique takes advantage of the fact that the original plan begins from a state that is equivalent to resetting all of the subsystems.

Thus, by using a combination of planning techniques and replanning methods, the Dplan planning system can generate antenna operation plans for the DSN. By automatically generating such plans, Dplan has both improved the reliability of the DSN and greatly reduced maintenance and operation costs.


### Acknowledgments

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract to the National Aeronautics and Space Administration. The Dplan task is part of the Network Automation work area of the DSN Technology Program, managed by Chad Edwards.

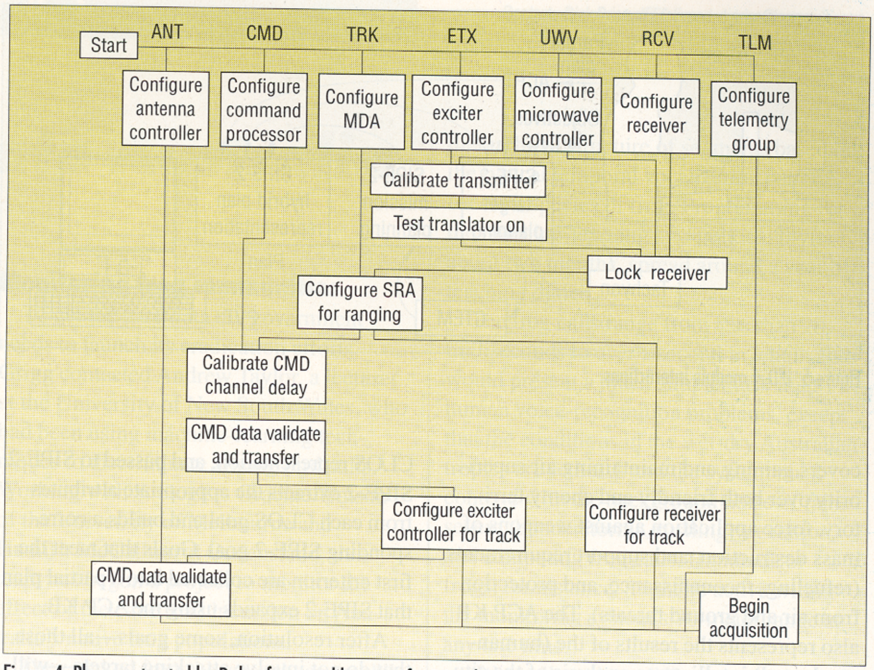




**Figure 4. Plan constructed by Dplan for precalibration of a 34-m beam waveguide antenna for telemetry, commanding, and ranging track.**

## Using SIPE-2 to integrate planning for military air campaigns

*Thomas J. Lee and David E. Wilkins, SRI International*

In 1995 and 1996, SRI participated in an integrated feasibility demonstration to show, in an operational environment, the relevance of generative planning in the domain of military air campaign planning. The demonstration took place in May and June 1996, for both DARPA and Rome Laboratory representatives and for representatives from Air Combat Command, Langley Air Force Base. The demonstration was well received, and as a result has become a cornerstone of ongoing work under the DARPA/Rome Labs Planning Initiative, the program under which it was funded.

SRI's SIPE-2 planning system was used to perform generative planning. As Figure 5 shows, SIPE-2 was integrated with three other systems—ACPT, an air campaign plan authoring tool developed at ISX; CTEM, a force requirements estimator and scheduler developed at AEM Services; and the Plan Visualization Tool (PVT), developed by General Electric and based on its Tachyon system. The role of SIPE-2 was to accept a partial plan created with ACPT, expand the unsolved goals in the plan by using the knowledge encapsulated in SIPE-2's air campaign planning knowledge base (ACP KB) and supplemented by recommendations made by CTEM, and pass the resulting plan to PVT for inspection.

The resulting integrated system provides several capabilities to an Air Force staff planner: a feasibility analysis in terms of resources and time required to execute the plan; a visualization of the plan, including resource and schedule shortfalls, in an easily understandable form; and an environment for plan modification.

### The planning problem

The problem confronting an air campaign planner is complex. Given a set of high-level political and military goals (for example, "Protect US citizens and forces from hostile attack"), the planner refines the goals that are attainable (wholly or in part) by the employment of air power into more specific goals. This process iterates until each goal is directly attainable by the execution of a *mission.* A group of identical aircraft acting in concert performs a mission. Each mission consists of a mission type, a time and place, a type of aircraft, and the number of sorties required to execute the mission. Thus, a mission might be expressed as "Four F-15Cs to escort strike package P to target T on day D+1." Low-level mission planning details, such as flight path and altitude profile, are outside this application's scope.

There are often multiple ways to refine goals into subgoals. These refinements reflect the different strategies and tactics that are available. For example, a refinement of the preceding goal might include a subgoal to defend a friendly country F that is near a belligerent nation. Further refinements might contain defensive tactics ("Patrol the borders of F"), preemptive tactics ("Attack hostile airbase AB near F"), or a combination thereof. Available options are constrained by the situation, which includes local geography, the enemy's characteristics and capabilities, restrictions imposed by political authority, and the availability of aircraft and other assets.

These strategies, tactics, and constraints are represented in the ACP KB, developed in consultation with the Checkmate office of the US Air Force. The knowledge base

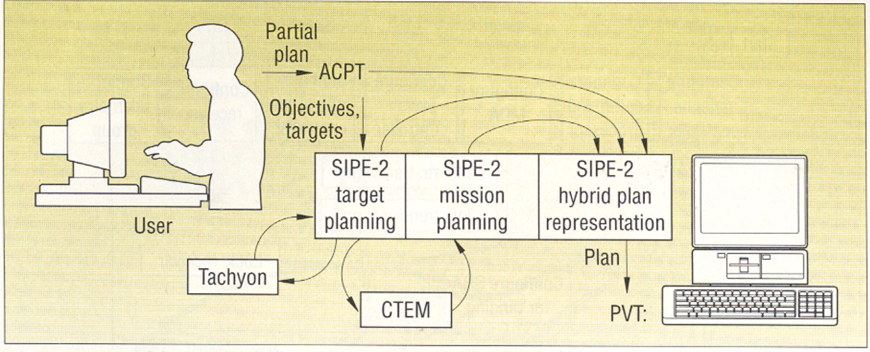



Figure 5. IFD-4 module interactions.

covers gaining and maintaining air superiority over both friendly and enemy territory, force application against weapons of mass destruction, and support requirements (refueling, reconnaissance, and protection from air and ground threats). The ACP KB also represents the results of the (human-conducted) intelligence analysis of the situation, which serves to focus the generative planner on enemy strengths, weaknesses, and other salient aspects of the situation.

## The planning technology

The earlier essay, "Using SIPE-2 to plan emergency response to marine oil spills," describes SIPE-2. The plans generated for this demonstration are the largest ever generated by SIPE-2, containing several thousand primitive actions. We accomplished this application with only one small change to SIPE-2, purely for efficiency reasons. In a low-level routine, SIPE-2 had previously used an algorithm that was quadratic in the number of choice points in the plan. We replaced this algorithm with a linear algorithm (made possible by sorting the choice points).

## The application system

To understand the integrated system and its capabilities, it is useful to follow a plan as it is processed by the components of the system. A human operator interacts with ACPT to create the initial plan. The human planner successively decomposes and refines the plan's high-level political and military goals until they meet one of two criteria: the goal is resolved to the point at which the ACP KB has knowledge about how to solve it ("Achieve air superiority over friendly forces and enemy territory"), or the goal is resolved to a set of targets. The former goals pass to SIPE-2 for further refinement, and the latter pass to CTEM.

All goals are extracted from the ACPT plan, converted from the plan's object-oriented database representation into a CLOS representation, and passed to SIPE-2. SIPE-2 extracts the appropriate attributes from each CLOS goal and builds a corresponding SIPE-2 goal. Goals that meet the first criterion are collected into a partial plan that SIPE-2 expands using the ACP KB.

After resolution, some goals—all those that do not involve attacking targets—will have been resolved down to the primitive-action level; these require no further processing until subsequent scheduling and resource allocation. The remaining target goals—those generated by SIPE-2 as well as by the human planner—are collected and passed to CTEM. CTEM recommends the type and number of strike assets (either cruise missiles or a combination of aircraft and munitions), and schedules the strikes on the basis of goal priority and availability of strike assets. CTEM also recommends a grouping of strikes into *packages* of strikes for delivery more or less simultaneously.

Once CTEM has made strike recommendations, the next step is to determine the support requirements of each strike package. These requirements include defense against hostile aircraft and surface-to-air missiles, refueling, and reconnaissance. SIPE-2 generates the corresponding support missions as follows. The CTEM recommendations are input and converted into SIPE-2 support-package goals, which are then collected into a second (post-CTEM) SIPE-2 plan. SIPE-2 then solves these goals, using the ACP KB, which contains knowledge about alternatives for strike package support and the types and numbers of aircraft required.

At this point, all planning goals have been fully expanded into missions. To determine whether the campaign plan is feasible in terms of available resources, all missions are collected from both the pre-CTEM and post-CTEM SIPE-2 plans, and passed to a resource-allocation module. This module allocates resources to missions in a nonoptimized, greedy manner. If resources become exhausted, limited alternatives are explored, and resource shortfalls may be identified.

The completed campaign plan now consists of three components—the pre-CTEM and post-CTEM SIPE-2 plans, and the human-generated ACPT plan. These are linked by SIPE-2 into a unified view of the campaign plan by associating all goals with their subgoals that reside in another component.

SIPE-2 then writes this unified plan in the input syntax of PVT, for presentation to the human planner. PVT, which replaces the SIPE-2 graphical user interface, was specifically designed for air campaign planners. Missions with resource shortfalls are annotated so that they can be highlighted in the plan display. Furthermore, all goal linkages are conveyed; these linkages enable PVT to propagate resource shortfall information upwards through the goal ancestry, and allow the display of all high-level goals that may fail because of resource shortages detected at lower levels. Mission scheduling information also passes to PVT and propagates upwards. This information is used to display the duration of all goals and missions, and to identify schedule overruns.

PVT highlights problems in the plan. To fix problems, the system operator invokes a plan modification and control module to change the plan. The operator then uses SIPE-2 to expand the modified campaign plan, and SIPE-2 displays the revised plan in PVT. Currently, this application supports only the addition of goals and limited modifications of planning assumptions and of available resources. However, SIPE-2 can handle other plan modifications.

In summary, the integrated system provides plan visualization and feasibility estimation, as well as plan modification to fix detected problems.

## Acknowledgments

This effort was sponsored by the Air Force Materiel Command and the Defense Advanced Research Projects Agency.

## Reference

1. D.E. Wilkins and R.V. Desimone, "Applying an AI Planner to Military Operations Planning," *Intelligent Scheduling*, M. Zweben and M. Fox, eds., Morgan Kaufmann, San Francisco, 1994, pp. 685–709.