RL-TR-96-210
Final Technical Report
January 1997

DISTRIBUTED, INTERACTIVE DEVELOPMENT AND MONITORING OF TRANSPORTATION PLANS IN DYNAMIC ENVIRONMENTS

University of Pittsburgh

Jointly Sponsored by
Rome Laboratory and
Advanced Research Projects Agency
ARPA Order No. A008

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-210 has been reviewed and is approved for publication.

APPROVED:

NORTHRUP FOWLER III
Project Engineer

FOR THE COMMANDER:

JOHN A. GRANIERO
Chief Scientist
Command, Control, & Communications Directorate

DISTRIBUTED, INTERACTIVE DEVELOPMENT AND MONITORING OF TRANSPORTATION PLANS IN DYNAMIC ENVIRONMENTS

Contractor: University of Pittsburgh
Contract Number: F30602-93-C-0038
Effective Date of Contract: 18 February 1993
Contract Expiration Date: 30 June 1996
Program Code Number: 3D30
Short Title of Work: Distributed, Interactive Development and Monitoring of Transportation Plans in Dynamic Environments
Period of Work Covered: Feb 93 - Jun 96

Principal Investigator: Martha E. Pollack
Phone: (412) 624-9590

RL Project Engineer: Northrup Fowler III
Phone: (315) 330-3011

Approved for Public Release; Distribution Unlimited

This research was jointly sponsored by the Advanced Research Projects Agency of the Department of Defense and by Rome Laboratory, and was monitored by Northrup Fowler III, RL/C3C, 525 Brooks Road, Rome, NY 13441-4505.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1997	Final Feb 93 - Jun 96

4. TITLE AND SUBTITLE
DISTRIBUTED, INTERACTIVE DEVELOPMENT AND MONITORING OF TRANSPORTATION PLANS IN DYNAMIC ENVIRONMENTS

5. FUNDING NUMBERS
C - F30602-93-C-0038
PE - 61101E & 62702F
PR - A008
TA - 00
WU - 01

6. AUTHOR(S)
Martha E. Pollack

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Advanced Research Projects Agency
3701 Fairfax Drive
Arlington VA 22203-1714

Rome Laboratory/C3C
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-96-210

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Northrup Fowler III/C3C/(315) 330-3011

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This project was aimed at adapting relevant techniques developed in real-time and distributed operating systems, and combining them with techniques from real-time and distributed AI, to support plan generation in dynamic, multi-agent environments, including crisis-action settings. We describe DIPART - The Distributed, Interactive Planner's Assistant for Real-time Transporation planning - a prototype simulation system. We also describe advances made in the project in three areas (i) planning technology, (ii) meta-level reasoning to control planning, and (iii) coordination of distributed planning.

14. SUBJECT TERMS
Planning, Meta-Reasoning, Distributed Planning

15. NUMBER OF PAGES
52

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

1 Executive Summary

This document summarizes the research conducted between Feb. 18, 1993 and June 30, 1995 on the project "Distributed, Interactive Development and Monitoring of Transportation Plans in Dynamic Environments", supported by the Rome Laboratory (RL) of the Air Force Material Command and the Defense Advanced Research Projects Agency (Contract F30602-93-C-0038). The project was aimed at adapting relevant techniques developed in real-time and distributed operating systems, and combining them with techniques from real-time and distributed AI, to support plan generation in dynamic, multi-agent environments. In support of this goal, we built DIPART—the Distributed, Interactive Planner's Assistant for Real-time Transportation planning—a prototype simulation system that includes a network of agents, each of which assists a human planner, and a simulated dynamic environment, which implements Reece and Tate's Pacifica NEO scenario [38]. The key accomplishments of this project fall into four main categories:

- The design and development of the DIPART testbed system (see Section 2.)
- Advances in planning technology (see Section 3), including the development of
  - the LCFR strategy for efficient search control during plan generation;
  - an effective algorithm for generating plans in domains in which actions have costs associated with them, and
  - a new planning framework based on the use of constraint-satisfaction processing methods.
- Advances in meta-level reasoning to control the planning process in dynamic environments (see Section 4), including
  - the identification of relevant process-scheduling algorithms and their adaptation to the control of planning in dynamic environments, and
  - the development of a framework for reasoning about which contingencies to plan for first, when time is limited.
- Advances in coordinating a distributed planning process in a multi-agent setting (see Section 5), including the development of
  - software support for flexible communication strategies among planning agents;
  - a load balancing mechanism for distributing the planning effort effectively;
  - algorithms for merging individually formed plans, and
  - a method for efficiently coordinating the planned actions of agents with minimal communication.

2 The DIPART System

Many current and potential AI applications are intended to operate in dynamic environments, including those with multiple agents. An important example is crisis action planning, which is typically a distributed process, involving multiple planners each tasked with forming plans to meet some subset of the overall mission objectives. During planning, changes that occur in the world can affect the quality of the plans being created. When planning and execution are interleaved, as they often must be in crisis situations, changes can also affect the quality of plans whose execution has already begun. To operate in such environments, standard AI plan-generation technology must be augmented with mechanisms for managing changing information, for focusing attention when multiple events occur, and for coordinating with other planning processes. In the DIPART project, we have been concerned with the development and analysis of such techniques. Many of the techniques we have explored derive from theoretical work in real-time AI and in related fields, such as real-time operating systems.

To support our research on plan generation in dynamic, multi-agent environments, we built DIPART—the Distributed, Interactive Planner's Assistant for Real-time Transportation planning. DIPART is a prototype simulation system that includes a network of agents, each of which assists a human planner, and a simulated dynamic environment, which implements the Pacifica NEO scenario [38].

2.1 System Overview

The DIPART system consists of a network of communicating nodes each assisting a human planner, plus a simulated environment [46]. The underlying idea is that each planner has responsibility for forming and overseeing the execution of some set of plans that are carried out in the (simulated) environment. Each planner may have only a restricted view of the environment and of the activities of the other planners; although cooperation among the planners may be desirable, it is not automatic. Figure 1 illustrates the overall system architecture, highlighting the internal architecture of a single node. Because each node performs the role of an intelligent assistant, we sometimes refer to the nodes as "agent processes".

The internal architecture of each DIPART node is based on a generic model of process scheduling, similar to those found in the literature on operating systems [45]. Incoming messages are stored on a Message Queue (MQ), and indicate events that may require attention. Often the value of responding to a particular message is time dependent. Thus, a mechanism is needed to determine what processes should be invoked in response to each message, and to schedule the selected processes. In our model, the module that makes these decisions is the Locus of Meta-Level Control (LMC); it is responsible for invoking various (object-level) processes, which we call Reasoning Modules (RMs). The

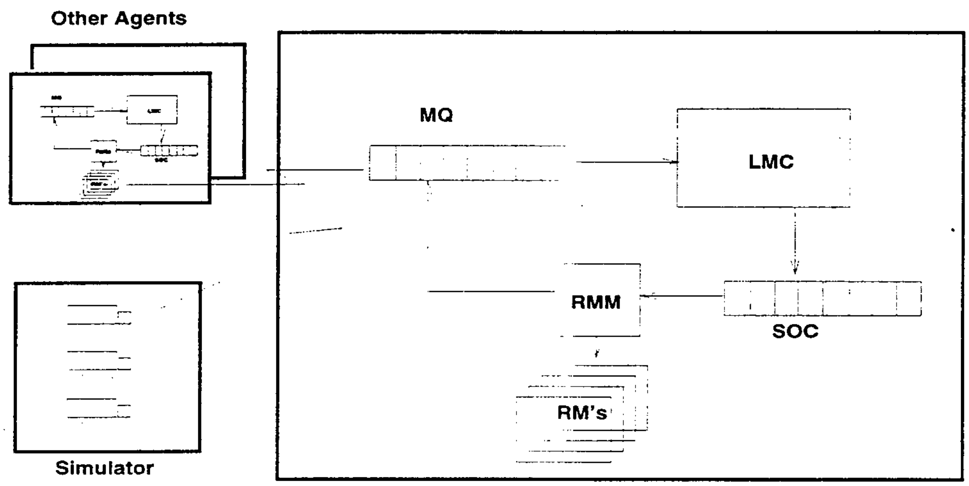

Figure 1: DIPART Architecture

RMs include a resource estimator, which estimates the amount of computational and external resources required for a given task, a planner, which computes plans to achieve specified goals, and an execution monitor, which tracks the performance of plans.

As shown in the figure, the LMC performs its task by posting entries in a Schedule of Computation (SOC). These entries include information about which RM to invoke, the input to that process, the invocation deadline (i.e., the time after which the system should no longer bother to invoke the process), and, in some circumstances, the amount of time to allocate to the process in question. A process controller, called the Reasoning Modules Manager (RMM) reads entries from the SOC and then invokes the appropriate process. Individual processes may also generate messages if follow-on computation is needed. A global database stores information that can be used by both the LMC and the object-level processes.

In addition to the agent nodes, DIPART includes a simulator which has been tailored to Pacifica NEO scenario, described below. It runs as a separate process in the overall DIPART system. It represents the "actual" state of the world; in contrast, the models of the world kept by individuals agents may be limited or may become out-of-date, as they are intended to represent the views that the agents currently have, given the information they have so far received. The simulator is designed to allow modeling of resource allocation to agents.

The DIPART system has been implemented on DECStation 5000 workstations, under Ultrix 4.3, using Allegro Common Lisp and the Garnet interface-development system. Each of the agent nodes runs on its own processor, as does the simulated environment. Within each node, the LMC runs in one thread, and the RMs in another. A communication package based on UDP has also been implemented to support inter-node communication.

2.2 The Pacifica Scenario

To ground our research, we employ the Pacifica NEO scenario, developed by Reece and Tate for the RL/ARPA Planning Initiative as part of the PRECiS environment [38]. This scenario involves the fictional island nation of Pacifica, on which a number of U.S. citizens are located. The island has various natural and man-made features, including cities, an airport, bridges, roads, and a volcano. Because of an expected uprising, the citizens need to be evacuated. For this, they must first be brought by truck to the capital city, where the airport is located. Evacuation can be complicated by unexpected road or bridge closings, either as a result of natural forces, e.g., a volcano, or hostile human forces; it can also be complicated by the fact that the citizens may be scattered around the island, and must themselves get to major cities before being taken by truck to the capital.

We assume that the NEO is to be planned and overseen by several human planners (typically, we run DIPART with between 2 and 6 planning nodes). Each human planner is responsible for a different component of the operation; although the task may be divided in various ways, we generally assign each planner the task of moving citizens from one city to the capital. The exact number of citizens and their current location may not be fully known to each planner. Each human planner is assisted by a DIPART node; the human submits goals to the node, and can query the node for current status information. The nodes are then responsible for forming plans to satisfy the user's goals, for coordinating communication with other planners, and for alerting the user to reports from agents in the (simulated) world.

3 Advances in Planning Technology

The key task performed by DIPART nodes is plan generation: human users input goals, such as evacuating a certain number of citizens from some city, and the DIPART node generates, dispatches, and monitors the execution of a plan to carry out that goal. Consequently, a central focus of our research has concerned the development of efficient planning algorithms.

3.1 Control during Planning

Many current state-of-the-art planners make use of partial-order causal link (POCL) algorithms [27, 30]. POCL planning involves searching through a space of partial plans, where the successors of a node representing partial plan P are defined to be the refinments of P. As with any search process, POCL planning requires effective control; in POCL planning, search control has two components. The first, *node selection,* involves choosing which partial plan to refine next. Most POCL algorithms use best-first search to perform node selection. Once a partial plan has been selected, the planner must then perform *flaw selection,* which involves choosing either a threat to resolve (typically, by promotion, demotion, or separation) or an open condition to establish (by adding a new step to the plan or adding a new causal link to an existing step). Unless it is impossible to repair the selected flaw, new nodes representing the possible repairs are added to the search space.

In [19], we explored a flaw-selection strategy, the Least-Cost Flaw Repair (LCFR) strategy, which can be seen as a generalization of the DUnf strategy that had been proposed by Peot and Smith [32]. In LCFR, we define the *repair cost* of any flaw—either threat or open condition—to be the number of nodes generated as possible repairs. LCFR is the strategy of always selecting a flaw with the lowest possible repair cost at a given node. LCFR will delay any threat that is unforced (repair cost > 1) in favor of a threat that is forced (repair cost $<= 1$.) By treating all flaws uniformly, LCFR also applies a similar strategy to open conditions, preferring to handle open conditions that are forced over open conditions, or threats, that are not. Similarly, LCFR handles the case in which all that remain are unforced threats: the LCFR strategy will select a threat with minimal repair cost.

Our experimental assessment of LCFR demonstrated that the power of DUnf does not come from delaying threat repairs *per se*, but rather from that fact that this delay has the effect of imposing a partial preference for least-cost flaw selection. Our experiments also showed that extending this to a complete preference for least-cost selection, as in LCFR, reduces search-space size even further. Details of the experiments can be found in [19]. Here we simply present the results of a key experiment, in which we compared 5 search strategies on 49 test problems from a variety of domains. Figure 2 plots the percentage

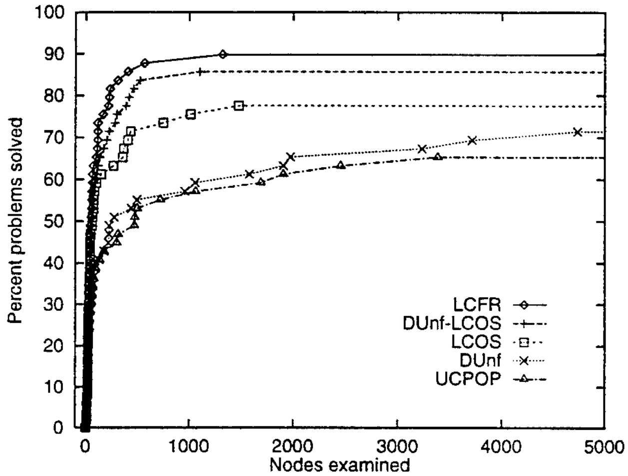

Figure 2: Comparison of planner search spaces

of test problems solved by each planner with a fixed number of nodes examined. (Each point $\langle x, y \rangle$ denotes that y% of the 49 test problems were solved by examining no more than x nodes.) As can be seen, the LCFR-based planner outperforms any of the others, including the two based on Peot and Smith's DUnf strategy.

As might be expected, the benefit of the LCFR strategy is not without a cost: specifically, performing least-cost flaw selection can incur a significant computational overhead. We therefore developed QLCFR, which reduces this overhead by approximating repair costs, and we demonstrated its effectiveness experimentally. Again, complete details can be found in [19]; subsequent work that builds on the LCFR approach includes [44, 43].

3.2 Cost-Directed Planning

The LCFR strategy described above is quite effective for planning problems in which alternative solutions to a planning problem are considered to be roughly equal—an assumption that is, in fact, made in much of the plan generation literature. In many domains, however, this assumption is not warranted: for any given planning problem, some solutions have lower execution cost, some are more likely to succeed, and so on. To handle such cases, we developed a "cost-directed" heuristic planner, which is capable of finding low-cost plans. The algorithm performs POCL planning, using an A^* strategy for node selection. The heuristic evaluation function is computed by a deep lookahead that calculates the cost of complete plans for a set of pre-defined top-level subgoals, under the (generally false) assumption that those subgoals do not interact. In our work so far, we have assumed that flaw selection is performed randomly, leaving it to future work to explore the question of which flaw selection strategies can best be integrated into our approach.

In [9], we show that the cost-directed planning algorithm not only leads to finding lower-cost plans, but in many circumstances it does this without negatively impacting the efficiency of planning: in fact, it can lead to a significant decrease in total planning time. This result is due in part to the fact that generating plans for a set of independent subgoals is exponentially less costly than generating a complete plan taking interactions into account[23]. At least in the limit, the cost of forming plans for subgoals treated independently does not significantly effect the computational complexity of the complete planning problem. Moreover, while focusing on lower-cost plans, the heuristic function effectively prunes the search space. Thus, the use of the deep evaluation in node selection can outweigh the marginal additional complexity. Our experiments demonstrate that the advantages of cost-directed planning increase with the complexity of the planning problem, where this is measured in terms of the amount of subgoal interdependence, the heterogeneity of the cost of actions, the average branching factor, and the number of subgoals and length of the minimal-cost plan.

3.3 Constraint-Based Planning

Both of the strategies for controlling planning described above build directly on a traditional POCL style of planning. To a large extent, POCL planning was originally motivated by an observation of the advantages of taking a "least-commitment" approach to planning. Least-commitment planning involves postponing decisions until constraints force them to be made. Any decision made when it is not forced is an "early commitment." POCL planning, as opposed to the earlier, state-spaced planning, made it possible to take a least-commitment approach to some decisions, particularly to the ordering of plan steps. However, POCL planners continue to rely to some degree on early commitments for other decisions, including variable binding, threat resolution, and choice of an operator to satisfy open conditions.

Because the least-commitment approach has, by and large, been successful where it has been tried, an obvious question is whether the least-commitment approach should be applied to *every* planning decision; in other words, is early commitment ever a good idea? An obstacle to addressing this question experimentally arises from the way in which POCL planners manage decision-making. They take what we call a *passive postponement* approach, choosing one decision at a time to focus on, and keeping all the other, postponed decisions (about how to achieve certain goals and how to resolve threats) on an "agenda," where they play no role in the plan generation process until they are selected for consideration. The items on the agenda may in fact impose constraints on the plan being generated, but these constraints are not available to the planning algorithm so long as the items remain on the agenda. The fact that constraints exist but are not always accessible makes it difficult if not impossible for a POCL planner to be made more "least commitment". Postponing decisions until they are forced implies being able to recognize whether any decision is forced, and this in turn implies that all the constraints that might affect a decision must be available to (and must be used by) the planning algorithm.

In response to these difficulties, we developed a new approach to planning, called *active postponement*, in which even postponed decisions play a role by constraining the plan being generated. This technique has been implemented in the Descartes system. The key idea in Descartes is to transform planning problems into Constraint Satisfaction Problems (CSPs) which can then be solved by applying both planning and CSP techniques. In general, a planning problem cannot be transformed into a single static CSP, however; instead it must be transformed into a *dynamic* CSP to which new constraints and variables can be added during the solution process. The dynamic CSP is then solved by breaking it down into static CSPs, to which standard CSP techniques may be applied.

As with the approaches discussed above, we have conducted a number of experiments to explore the power of constraint-based planning. These experiments demonstrate that passive postponement—even "smart" passive postponement, using a selection strategy like LCFR—can result in significant performance penalties. Further experiments show that it is worthwhile to extend the least-commitment approach much further than has been done in prior work. These results also suggest, however, that there are some fundamental limits to the effectiveness of the least-commitment approach, and that sometimes early commitments can increase planning efficiency. We have proposed a principled approach to deciding when to make early commitments in planning, based on an analysis of the ongoing constraint processing: specifically, early commitment is needed when the planning process is forced to make what we call unrestricted expansions. Details of the constraint-based planning approach, its implementation in Descartes, and the results of experiments using Descartes can be found in [20, 21, 18].

4 Advances in Control of Reasoning

As noted above, planning in DIPART occurs in a dynamic environment; often, one planning problem will have to be interrupted so that attention can be given to another planning problem. A central focus of the DIPART project has thus been the development and assessment of alternative strategies for meta-level reasoning, i.e. deciding how to allocate computational resources. Within the DIPART system this task is performed by the LMC. The LMC must decide what to do from messages that can arrive from four different sources:

1. the human user, who posts a new goal to the system or tells the system a new fact.

2. other nodes, which may be seeking information, or may have information to share, or may have goals that they would prefer to be handled by someone else.

3. agents situated in the simulated world, who may transmit a message to their supervising agent (i.e. DIPART node) to report an unexpected change in the environment.

4. reasoning modules within the node itself, which post messages identifying information about tasks that are in need of further processing by other RMs.

4.1 Process Scheduling Algorithms for Meta-Level Control

The problem of allocating reasoning resources is sometimes called the *deliberation-scheduling problem*. Previous approaches to deliberation scheduling in AI include the use of off-line allocation of on-line deliberation time for tasks with known computational demands [14, 2, 48], and the application of decision-theoretic estimations of optimal computational sequences [41]. Heuristic strategies have been proposed as well [34].

The deliberation-scheduling problem bears a strong similarity to the problems of process scheduling in real-time operating systems [45], job scheduling in operations research [33], and transmission scheduling in local area networks [28]. Not all process- or job- or transmission-scheduling algorithms are applicable to deliberation scheduling, however. In particular, we require scheduling algorithms that are:

- *on-line*, i.e., construct schedules at run time;

- *dynamic*, i.e., support the random arrival of tasks;

- *stochastic*, i.e., support tasks with random computation times; and

- *soft real-time*, i.e., support the scheduling of tasks that yield less than maximal value if completed after some critical period.

Two simple and well-researched scheduling algorithms are Earliest Deadline First (EDF) and Least Slack First (LSF). These both incorporate deadline information and consequently achieve better results than algorithms that do not, such as First-in-first-out (FIFO) and Round-robin [17, 28]. It is known that, for a *schedulable* set of processes, i.e., one for which there exists an optimal schedule where all deadlines can be met, EDF and LSF produce a schedule that meets all deadlines, and hence performs optimally [1]. However, the performance of these two algorithms degrades sharply when the system is *saturated*, i.e., it has to deal with a non-schedulable set of tasks.

To schedule saturated job sets effectively, scheduling algorithms must take into account the cost of missing a deadline. This is particularly true when there are trade-offs in the acceptance rate and the deadline miss rate of tasks in the system. The environments of autonomous agents typically present such trade-offs: it may well be worth missing the deadlines for some tasks in order to achieve higher-quality performance on other tasks. Such trade-offs can be evaluated with the aid of *value-density* assessment tools. The value density of any task t is defined to be the value to the system of completing t divided by its remaining computation time. Value-density assessments are included in scheduling algorithms such as Best-Effort (BE) and Dynamic-priority (DP); previous research has shown that these algorithms perform better than EDF and LSF in saturated environments [17, 28].

We have explored the usefulness for deliberation scheduling of the value-density measure and the algorithms that rely on it. Specifically, we identified appropriate candidate algorithms, conducted preliminary experiments to compare their performance of these algorithms, demonstrated a proof-of-concept use of these algorithms for deliberation scheduling in the DIPART system, and analyzed current limitations of the proof-of-concept system, i.e., identified certain assumptions that are made in the existing algorithms that must be relaxed to support full-fledged deliberation scheduling. In addition, we developed a modification of the Best Effort algorithm that results in improved performance for the DIPART job mix. Details can be found in [39, 40].

4.2 Contingency Selection in Plan Generation

Classical AI plan generation systems assume static environments and omniscient agents, and thus ignore the possibility that events may occur in unexpected ways—that contingencies might arise—during plan execution. The plans these systems construct are simply sequences of actions: they assume that the state that will occur after action A_i is taken is the one in which A_{i+1} should be performed. The problem with classical planners is, of course, things do not always go "according to plan."

In contrast, universal planning systems [42] and more recent Markov-decision process-based systems (e.g., [7, 3]) make no such assumptions. They produce "plans" or "policies" that are functions from states to actions. After performing an action in one state, the agent then looks to see what new state it is in, and performs the action prescribed for that state. Effectively, every contingency is handled at planning time. The problem with universal planning and MDP systems is that the state space is typically much too large for them to be effective.

Conditional planners take the middle road. They do not assume that the result of each action taken is necessarily known at planning time; instead they allow for both conditional actions with multiple possible contingencies, and for sensing actions that allow agents to determine which contingency occurred (e.g., [8, 31, 37]) A key question in conditional planning is: how many, and which contingencies should be selected so that the plan can be extended to include actions that will be taken in case the contingency fails? One cannot plan for all possible failures, or one will inherit the problems of universal planners. One cannot ignore all possible failures, or one will inherit the problems of classical planners.

The need to select contingencies was observed by Peot and Smith [31, p. 196], who, in describing the conditional planner CNLP, noted that "the size and complexity of the plans generated by CNLP increase exponentially with the number of observation actions in the plan. The amount of computation may be reduced by attaching a relative likelihood measure to the various contexts in the plan ... and skip[ing] contexts that are sufficiently unlikely ...". We have developed a planning algorithm that generalizes this suggestion, by deciding which of the possible failures to plan for, based not just on their likelihood of occurrence, but also on the damage they would cause if in fact they were to occur. Put otherwise, our algorithm attempts to identify the influence each contingency would have on the outcome of plan execution, and then proceeds by giving priority to the contingencies whose failure would have the greatest negative impact.

We define contingencies to be alternative outcomes of probabilistic actions, and our approach is to directly reason about the value of planning for the failure of various contingencies. In order to make this feasible, we have adopted a different strategy towards conditional planning than that taken in some earlier systems, notably C-BURIDAN [8], which is, to our knowledge, the only partial-order conditional planner that does not necessarily plan for all contingencies. C-BURIDAN constructs branches for alternative contingencies in a somewhat indirect fashion. It sometimes discovers that there are two incompatible actions, say, $A1$ and $A2$, each of which can achieve some condition C. To resolve the threats between $A1$ and $A2$, C-BURIDAN introduces an observation action O with two contingent outcomes, and then "splits" the plan into two branches, associating each outcome with one of the actions. In the process of doing this, C-BURIDAN does not reason about whether both those contingent outcomes are worth planning for.

Our algorithm works by iterative refinement. It first finds a skeletal plan to achieve the goals, and then during each iteration selects a contingency whose failure will have a maximal *disutility*, i.e., one that will have a maximal negative impact if it does not occur. The algorithm then extends the plan to include actions to take in case the contingency fails. Iterations proceed until the expected utility of the plan exceeds some specified threshold. Details of this work can be found in [29].

5 Advances in Distributed Planning and Communication

One addition research topic that centrally concerns us is the specification of appropriate communication and coordination strategies for multi-agent, dynamic planning. We first briefly describe the communication package we have implemented to support communication among DIPART nodes, and then sketch approaches to multi-agent planning that we have been investigating.

5.1 Software Support for Communication

Communication among nodes in DIPART is built on a group management model [6]. Groups of processes (or, in our case, agents) cluster into a single logical entity such that all communications sent by a member of the group are received by all members of the group. Thus multi-process communication is achieved by a single operation rather than by a series of operations to a (potentially unknown or partially known) set of individual agents. Group operations can take advantage of network-multicast capability, thus reducing communication overhead and increasing concurrency.

Using a group management model, we have implemented a set of communication primitives that enable the basic group operations (e.g., form a group, dissolve a group, join a group, leave a group, invite to a group, and exclude from a group) and communication actions (e.g., send, send and block, receive, group-cast, group-cast and block, receive any, receive any and block). Groups may have different structures, which determine the relationship among group members. In a *coordinated group*, the owner of the group must approve any new members, while in *peer groups*, all new members are accepted. There are also different group types: in a private group, communication is restricted only to the members of the group, while non-members may send messages to public groups. We have also implemented a group server, which maintains information about the status and membership of each group, and is responsible for synchronizing group actions. Additional details can be found in [49, 24].

5.2 Load Balancing for Distributed Planning

The communications package can be used to support a process of load balancing among the DIPART agents, so that no agent falls behind as a result of having too many responsibilities, while other agents sit idle. We have investigated a range of load-balancing techniques developed in the distributed operating-systems literature, focusing in particular on those that use dynamic thresholds.

The purpose of dynamic thresholds is to give the agent more flexibility to adapt itself to a changing environment. Consider a gift shop as an example: if the shop receives a hundred customers in a regular week day it is a busy day. However, if the shop receives a hundred customers just before Christmas, it is not a busy day. Similarly, in a military application, one would expect higher processing loads during crisis situations than during routine, peacetime operations. Instead of determining *a priori* what is a high load, dynamic load balancing evaluates the load of an agent at running time according to the partial information it possesses about the environment. As a consequence, given the same amount of tasks to perform the same agent may consider itself highly loaded or lightly loaded depending on its estimation of the system load. Dynamic thresholds are suitable in dynamic environments when a system must avoid unnecessary communications that would add an extra overhead to an already overloaded system.

Another way to lessen communication load is to employ selective unicasting. In load balancing, one faces a trade-off between the cost of exchanging messages and the necessity of having an information accurate enough to provide efficient load balancing. When initiating load balancing an agent could send a message to every other agent asking information about their loads and wait for the answers before selecting the best agent to balance with. However this classical scheme has two drawbacks: it requires many messages to be exchanged, and it is not fault-tolerant—it does not include provisions for the case in which one or more agents is unable to respond. In contrast, in selective unicasting the messages concerning the exchange of information about the load of the system are piggybacked to task balancing messages and therefore induce almost no overhead. Also, the scheme is non-blocking, and will not collapse should one or more agents fail. However, selective unicasting with piggybacking results in the agent being unable to access complete information about the system's current load. Thus, each agent must estimate this information by using data about the previous load history.

The load-balancing algorithms we constructed were implemented and subjected to experimentation to assess their performance relative to a variety of alternatives, including a broadcasting scheme; we also studied the relative effectives of client-driven, server-driven, and hybrid variants. We measured two things: *throughput* and *efficiency*, both of which were defined in terms of the Pacifica scenario. For instance, throughput was taken to be the ratio p/d, where p is the number of passengers for whom transportation has been requested, and d is the delay between the time of the first goal submitted and the completion of the last request. The throughput is given by the ratio p/d.

Details of the experiments, and complete results, can be found in [25, 26]. The most important result is that selective load balancing (hybrid) yields a good throughput, even compared with load balancing with broadcast. When compared with the lower bound, hybrid load balancing achieves a performance 34% higher, client driven load balancing achieves a performance 15% higher, and server driven load balancing achieves a performance 11% higher. Compared to the upper bound, selective load balancing performs only 7% worse, while using many fewer messages. Thus it appears possible to achieve effective load balancing by using dynamic thresholds, even if communication must be minimized.

5.3 Plan Merging

The load-balancing work involves agents sharing the work, but each individually forming their own, more or less complete plans. Sometimes this is feasible, but at other times, agents need to form partial plans, which are then merged together. We identified four different types of situations in which some merging may be needed. In the first, a group of agents has to cooperatively achieve one common global goal. In the second type of situation, due to time constraints, execution of the plan is interleaved with the planning process itself. In the third third, each agent has its own private, individual goal. There is also a fourth situation, in which planning and execution are interleaved for a group of agents with private goals. The DIPART scenario can be viewed either as an instance of the second of the fourth type, depending on how much knowledge each of the human planning agents has about the plan for the overall mission.

For each of these situations, we described how a global plan is constructed through the process of incrementally merging sub-plans. By making use of the computational power of multiple agents working in parallel, the process is able to reduce the total elapsed time for planning as compared to a central planner. For the case in which agents do not have complete knowledge of the overall mission, we show how agents can reach consensus about what multi-agent plan to carry out using a voting procedure, without having to reveal full goals and preferences (unless that is actually necessary for consensus to be reached). Our technique also does away with the need to generate final alternatives ahead of time (instead, candidate states arise at each step as a natural consequence of the emerging plan). The agents iteratively converge to a plan that brings the group to a state maximizing the overall system utility. Details and experimental results can be in [11, 10, 13].

5.4 Multi-Agent Filtering

In addition to plan merging, which involves explicit coordination among agents, it is sometimes useful for agents to have a means of achieving coordination implicitly. We have been investigating a strategy for implicit coordination called *multi-agent filtering*. It extends a single-agent strategy, filtering, which was developed as a way of controlling reasoning in dynamic environments. The notion of single-agent filtering derives from the work of Bratman [4]; it involves an agent committing to the goals it has already adopted, and tending to bypass (or "filter out") new options that would conflict with their successful completion [5, 34, 35]. We and others have studied the effectiveness of filtering in domains with various characteristics[36, 22, 35].

Where single-agent filtering means tending to bypass options that are incompatible with an agent's *own* goals, multi-agent filtering means tending to bypass options that are incompatible with *any* agent's known or presumed goals. We examined several forms of multi-agent filtering, which range from purely implicit, in which agents have rules of legal action that lead to their avoiding conflict without ever reasoning explicitly about one another's goals, to minimally explicit, in which agents perform very shallow reasoning to assess whether their actions are incompatible with the likely intended actions of other agents. In no cases do the agents engage in any explicit negotiation.

Our experimental results on the efficacy of multi-agent filtering are presented in [12]. The most interesting and surprising result is that, at least for the simple, abstract environments so far bstudied, multi-agent filtering is a dominant strategy: no matter what proportion of the agents in some environment choose not to filter, those that do filter perform better.

6 Conclusions

The traditional planning paradigm in the AI literature has made a number of extremely strong assumptions, including the following:

1. There is a single planning agent, who is the only cause of change in the environment.

2. The planning agent knows all the relevant facts about its environment.

3. The goals presented to the agent remain unchanged throughout the process of planning and execution.

4. The goals are categorical, i.e., they are fully achieved or not (there is no notion of partial satisfaction).

5. The actions to be performed have certain outcomes.

Of course, these assumptions are violated in most real-world applications. In a crisis-action setting, for example, there will typically be many planning agents, each with incomplete information about the environment, who must coordinate with one another in the formation and execution of plans. The world will change not only as a result of these agents' actions, but also as a result of the actions of other agents, who may be hostile; changes may also occur independent of any agent's actions, for example, as a result of weather or other natural activity. As the situation develops, new goals may arise, and previous goals may become irrelevant. The actions to be performed will have uncertain outcomes, and it will be possible to achieve some goals only partially.

The DIPART project has been concerned with expanding the planning process, to weaken the first three assumptions. (We have been less concerned with the fourth and fifth assumptions; but see the growing body of work on decision-theoretic planning, e.g. [15, 47, 16].) We have made progress towards our goal of supporting automated planning in dynamic, multi-agent settings, by extending our understanding of the algorithms that can support planning. Thus, as described in this report, we have developed better, more efficient techniques for plan generation, for controlling the planning process, and for coordinating planning processes among multiple agents. In developing these techniques, we drew on earlier research both from real-time and distributed AI, and from real-time and distributed operating systems. Moreover, we employed an experimental methodology, in which we explored the performance of our proposed algorithms under different conditions by conducting systematic experiments using experimental platforms, including the DIPART simulator.

Of course, many research questions remain unanswered. In our view, perhaps the most important involves further development of the foundations of dynamic planning:

planning in situations in which one's goals and knowledge change, sometimes dramatically, over time. In particular, little has yet been done on the question of exploiting one's expectations about future change during the planning process. For example, if a planning agent expects that it will soon gain a large amount of additional information, it may choose to defer planning and/or to form only highly abstract plans until that additional information is received. A related problem involves forming plans that are robust with respect to anticipated changes in the environment. A plan that would be optimal if the environment does not change might well be suboptimal in light of changes that can reasonably be expected to occur. Current planning technology does not include mechanims for reasoning about expected change.

Another critical question still to explored is the cost of learning in dynamic environments. Just as there is a trade-off between communicating, to receive more information, and acting on the basis of what is already known, in a dynamic environment, an agent must weigh the cost of learning a new procedure for achieving a goal against the continued use of an already known, but possibly less efficient procedure. Complete deliberation about the utility of learning is not always feasible in a dynamic environment; more efficient, if suboptimal, strategies for determining when to attempt to learn must be designed. The development of such strategies awaits further investigation.

References

[1] K. Baker. *Introduction to Sequencing and Scheduling.* John Wiley, N.Y., 1974.

[2] M. Boddy and T. Dean. Solving time-dependent planning problems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 979–984, Detroit, MI, 1989.

[3] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the 14th Joint International Conference on Artificial Intelligence (IJCAI)*, 1995.

[4] M. E. Bratman. *Intention, Plans and Practical Reason.* Harvard University Press, Cambridge, MA, 1987.

[5] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[6] D. R. Cheriton and W. Zwaenpoel. Distributed process groups in the V kernel. *ACM Transactions on Computer Systems*, pages 77–107, 1985.

[7] T. Dean, L. P. Kaelbling, J. Kirman, and A. Nicholson. Planning with deadlines in stochastic domains. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 574–579, Washington, D.C., 1993.

[8] D. Draper, S. Hanks, and D. Weld. Probabilistic planning with information gathering. In *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems*, pages 31–36, 1994.

[9] E. Ephrati, M. E. Pollack, and M. Milshtein. A cost-directed planner: Preliminary report. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, 1996.

[10] E. Ephrati, M. E. Pollack, and J. S. Rosenschein. Exploitation of decision theory techniques in multi-agent planning. In *1994 AAAI Spring Symposium on Decision Theoretic Planning.* AAAI Press, 1994.

[11] E. Ephrati, M. E. Pollack, and J. S. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. In *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS)*, 1995.

[12] E. Ephrati, M. E. Pollack, and S. Ur. Deriving multi-agent coordination through filtering strategies. In *Proceedings of the 14th Joint International Conference on Artificial Intelligence (IJCAI)*, 1995.

[13] E. Ephrati and J. S. Rosenschein. Divide and conquer in multi-agent systems. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994.

[14] L. Greenwald and T. Dean. Solving time-critical decision-making problems with predictable computational demands. In *Proceedings of the Second International Conference on AI Planning Systems (AIPS)*, pages 25–30, Chicago, IL, 1994.

[15] P. Haddawy and M. Suwandi. Decision-theoretic refinement planning using inheritance abstraction. In *Proceedings of the Second International Conference on AI Planning Systems*, pages 266–271, Chicago, IL, 1994.

[16] S. Hanks, S. Russell, and e. Michael Wellman. *AAAI Spring Symposium on Decision-Theoretic Planning.* AAAI Press, Menlo Park, CA, 1994.

[17] E. D. Jensen, C. D. Locke, and H. Tokuda. A time-driven scheduling model for real-time operating systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, pages 112–122, 1985.

[18] D. Joslin. *Passive and Active Decision Postponement in Plan Generation.* PhD thesis, Intelligent Systems Program, University of Pittsburgh, 1996.

[19] D. Joslin and M. E. Pollack. Least-cost flaw repair: A plan refinement strategy for partial-order planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI)*, pages 1004–1009, Seattle, WA, 1994.

[20] D. Joslin and M. E. Pollack. Active and passive postponement of decisions in plan generation. In *Proceedings of the Third European Workshop on Planning*, 1995.

[21] D. Joslin and M. E. Pollack. Is 'early commitment' in plan generation ever a good idea? In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, Portland, OR, 1996. To appear.

[22] D. N. Kinny and M. P. Georgeff. Commitment and effectiveness of situated agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 82–88, Sydney, Australia, 1991.

[23] R. E. Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33:65–88, 1987.

[24] S. Lauzac. The DIPART communications package. Technical Report 93-24, University of Pittsburgh, Pittsburgh, PA, 1993.

[25] S. Lauzac. Load balancing algorithms for cooperative planning. Technical Report 95-26, Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, 1994.

[26] S. Lauzac and T. Znati. Comparative evaulation of cooperative plan execution strategies in multiagent environments. In *Proceedings of the 28th Annual Simulation Symposium*, 1995.

[27] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 634–639, Anaheim, CA, 1991.

[28] M. M. Nassehi and F. A. Tobagi. Transmission scheduling policies and their implementation in integrated-services high-speed local area networks. In *he Fifth Annual European Fibre Optic Communications and Local Area Networks Exposition*, pages 185–192, Basel, Switzerland, 1987.

[29] N. Onder and M. E. Pollack. Contingency selection for plan generation. In *AAAI Fall Symposium on Plan Execution*, 1996.

[30] J. S. Penberthy and D. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, pages 103–114, Cambridge, MA, 1992.

[31] M. Peot and D. E. Smith. Conditional nonlinear planning. In *Proceedings of the First International Conference on AI Planning Systems (AIPS-92)*, pages 189–197, College Park, MD, 1992.

[32] M. Peot and D. E. Smith. Threat-removal strategies for partial-order planning. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 492–499, Washington, D.C., 1993.

[33] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Prentice Hall, Englewood Cliffs, NJ 07632, 1995.

[34] M. E. Pollack. The uses of plans. *Artificial Intelligence*, 57:43–68, 1992.

[35] M. E. Pollack, D. Joslin, A. Nunes, S. Ur, and E. Ephrati. Experimental investigation of an agent-commitment strategy. Technical Report 94-31, Univ. of Pittsburgh Dept. of Computer Science, Pittsburgh, PA, 1994.

[36] M. E. Pollack and M. Ringuette. Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 183–189, Boston, MA, 1990.

[37] L. Pryor and G. Collins. Cassandra: Planning for contingencies. Technical Report 41, Institute for the Learning Sciences, Northwestern University, 1993.

[38] G. A. Reece, A. Tate, D. I. Brown, and M. Hoffman. The PRECiS environment. Technical report, ARPA-RL/CPE, 1993.

[39] Y. Ronen. Meta-level deliberation as scheduling: The use of operating-systems and operations-research techniques in meta-level control. M.S. Thesis, Intelligent Systems Program, University of Pittsburgh, 1995.

[40] Y. Ronen, D. Mossé, and M. E. Pollack. Value-density algorithms for the deliberation-scheduling algorithm. *SIGART Bulletin*, 7(2), 1996.

[41] S. J. Russell and E. H. Wefald. *Do the Right Thing.* MIT Press, Cambridge, MA, 1991.

[42] M. J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1039–1046, 1987.

[43] L. Schubert and A. Gerevini. Accelerating partial order planners by improving plan and goal choices. In *Proceedings of the International Conference on Tools with AI (ICTAI '95)*, 1995.

[44] R. Srinivasan and A. E. Howe. Comparison of methods for improving search efficiency in a partial-order planner. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1620–1626, 1995.

[45] A. S. Tannenbaum. *Modern Operating Systems.* Prentice Hall, Englewood Cliffs, 1992.

[46] S. Ur and Y. Ronen. Dipart user's manual. Technical report, University of Pittsburgh Department of Computer Science, 1995.

[47] M. Williamson and S. Hanks. Optimal planning with a goal-directed utility model. In *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, pages 176–181, 1994.

[48] S. Zilberstein and S. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 79(2), 1996.

[49] T. Znati and M. E. Pollack. DIPART, an interactive simulation platform for plan development and monitoring in dynamic environments. In *Proceedings of the 27th Annual Simulation Society.* IEEE Computer Society Press, 1994.

7 Project-Related Publications

The research described above resulted in the following publications:

1. R. Conticello. *Implementation of a User Interface for the DIPART System.* M.S. Project Report, Dept. of Computer Science, University of Pittsburgh, February, 1995. (Available as Tech. Report 95-27).

2. E. Ephrati, M. Perry, and J. S. Rosenschein. Plan execution motivation in multi-agent systems. *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems (AIPS)*, June, 1994.

3. E. Ephrati, M. Perry, and J. S. Rosenschein. Incentive to Work: Deriving Cooperation among Self-Interested Agents (Preliminary Report). *1994 AAAI Spring Symposium on Decision Theoretic Planning*, AAAI Press, March, 1994.

4. E. Ephrati, M. E. Pollack, and M. Milshtein. A Cost-Directed Planner: Preliminary Report. *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, August 1996.

5. E. Ephrati, M. E. Pollack, and J. S. Rosenschein. Exploitation of decision theory techniques in multi-agent planning. In *1994 AAAI Spring Symposium on Decision Theoretic Planning*, AAAI Press.

6. E. Ephrati, M. E. Pollack, and J. S. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS)*, June, 1995.

7. E. Ephrati, M. E. Pollack, and S. Ur. Deriving multi-agent coordination through filtering strategies. *Proceedings of the 14th Joint International Conference on Artificial Intelligence (IJCAI)*, August, 1995.

8. E. Ephrati and J. S. Rosenschein. Divide and conquer in multi-agent systems. *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, August 1994.

9. E. Ephrati and J. S. Rosenschein. Deriving consensus in multi-agent systems. To appear in *Artificial Intelligence.*

10. E. Ephrati and J. S. Rosenschein. A heuristic technique for multiagent planning. To appear in *Annals of Mathematics and Artificial Intelligence.*

11. E. Ephrati, G. Zlotkin, and J. S. Rosenschein. Meet your destiny: a non-manipulable meeting schedule. *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, October, 1994.

12. B. Field and T. Znati. α-channel: A network framework to support real-time performance guarantees. *IEEE Journal on Selected Area of Communications*, 11(8):1317-1329, 1993.

13. S. Hanks, M. E. Pollack, and P. R. Cohen. Benchmarks, testbeds, controlled experimentation, and the design of agent architectures. *AI Magazine*, 14(4):17–42, 1993.

14. D. Joslin and M. E. Pollack. Is 'early commitment' in plan generation ever a good idea? *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, August 1996.

15. D. Joslin and M. E. Pollack. Active and passive decision postponement in plan generation. *Proceedings of the 3rd European Workshop on Planning*, Sept. 1995.

16. D. Joslin and M. E. Pollack. Least-cost flaw repair: A plan refinement strategy for partial-order planning. *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, August 1994.

17. S. Lauzac. *Load balancing algorithms for cooperative planning.* M.S. Thesis, Dept. of Computer Science, University of Pittsburgh, January, 1995. (Available as Tech. Report 95-26).

18. S. Lauzac. The DIPART communications package. Technical Report 93-24, University of Pittsburgh, Pittsburgh, PA, 1993.

19. S. Lauzac and T. Znati. Comparative evaluation of cooperative plan execution strategies in multiagent environments. *Proceedings of the 28th Annual Simulation Symposium*, March, 1995.

20. A. Nunes. *Towards a Machine IQ: Some perspectives on evaluation metrics for embedded agents with an application to agents designed for the Pacifica evaluation scenario.* M.S. Thesis, Dept. of Computer Science, University of Pittsburgh, May, 1994. (Available as Tech. Report 94-28.)

21. N. Onder. *What contingencies to plan for.* Ph.D. dissertation proposal, Dept. of Computer Science, University of Pittsburgh, March, 1995. (Available as Tech. Report 95-25).

22. N. Onder and M. E. Pollack. Contingency selection for plan generation. To appear in *AAAI Fall Symposium on Plan Execution*, Nov., 1996.

23. M. E. Pollack. Evaluating plans, planners, and planning systems. *SIGART Bulletin* 6(1), 1995.

24. M. E. Pollack *et al.* The DIPART Project: A Status Report. In *Proceedings of the ARPI Annual Meeting*, Tucson, AZ, Feb., 1994.

25. M. E. Pollack. Filtering as a reasoning-control strategy: An experimental assessment. In *Proceedings of SOAR '93 (Space Operations and Applications, and Research Symposium)*, Aug., 1993.

26. M. E. Pollack. Planning in dynamic environments: The DIPART system. In A. Tate, editor, *Advanced Planning Technology.* AAAI Press, Cambridge, MA 1996.

27. M. E. Pollack, D. Joslin, A. Nunes, S. Ur, and E. Ephrati. *Experimental Investigation of an Agent Commitment Strategy.* Dept. of Computer Science, Univ. of Pittsburgh, Tech. Report 94-31.

28. Y. Ronen, D. Mosse', and M. E. Pollack. Value-Density Algorithms for the Deliberation Scheduling Problem. *SIGART Bulletin*, 7(2), 1996.

29. Y. Ronen and M. E. Pollack. Value-Density Algorithms for Deliberation Scheduling. In *Proceedings of the IJCAI '95 Workshop on Deliberation Scheduling and Anytime Algorithms*, Aug. 1995.

30. R. Simon, T. Znati, and R. Sclabassi. DIPCS: An InterProcess Communication Architecture for Distributed Multimedia Systems. To appear in *International Journal of Multimedia Tools and Applications*

31. S. Ur and Y. Ronen. DIPART User's Manual. Department of Computer Science, University of Pittsburgh, 1995.

32. R. M. Young, J. D. Moore, and M. E. Pollack. Towards a principled representation of discourse plans. *Proceedings of the 16th Annual Meeting of the Cognitive Science Society*, July, 1994.

33. R. M. Young, M. E. Pollack, and J. D. Moore. Decomposition and causality in partial-order planning. *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems (AIPS)*, June, 1994.

34. T. Znati and M. E. Pollack. DIPART, an interactive simulation platform for plan development and monitoring in dynamic environments. *Proceedings of the 27th Annual Simulation Society.* IEEE Computer Society Press, April, 1994, pages 236-244.

35. T. Znati. A priority based protocol for transmission of real-time traffic. *ISMM International Journal of Computers*, 1995.

36. T. Znati and R. Melhem. A Uniform framework for dynamic load balancing strategies in distributed processing systems. *Journal of Parallel and Distributed Computing*, 2:246-255, 1994.

DISTRIBUTION LIST

addresses	number of copies
NORTHRUP FOWLER III ROME LABORATORY/C3C 525 BROOKS RD ROME NY 13441-4505	10
UNIVERSITY OF PITTSBURGH ATTN: MARTHA E. POLLACK DEPT OF COMPUTER SCIENCE PITTSBURGH PA 15260	5
ROME LABORATORY/SUL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ROME LABORATORY/C3AB 525 BROOKS RD ROME NY 13441-4505	1
ATTN: RAYMOND TADROS GIDEP P.O. BOX 8000 CORONA CA 91718-8000	1
AFIT ACADEMIC LIBRARY/LDEE 2950 P STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1

OL AL HSC/HRG, BLDG. 190 1
2698 G STREET
WRIGHT-PATTERSON AFB OH 45433-7604

US ARMY STRATEGIC DEFENSE COMMAND 1
CSSD-IM-PA
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

NAVAL AIR WARFARE CENTER 1
6000 E. 21ST STREET
INDIANAPOLIS IN 46219-2189

COMMANDER, TECHNICAL LIBRARY 1
474700D/C0223
NAVAIRWARCENWPNDIV
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6001

SPACE & NAVAL WARFARE SYSTEMS 2
COMMAND (PMW 178-1)
2451 CRYSTAL DRIVE
ARLINGTON VA 22245-5200

COMMANDER, SPACE & NAVAL WARFARE 1
SYSTEMS COMMAND (CODE 32)
2451 CRYSTAL DRIVE
ARLINGTON VA 22245-5200

CDR, US ARMY MISSILE COMMAND 2
RSIC, BLDG. 4484
AMSMI-RD-CS-R, DOCS
REDSTONE ARSENAL AL 35898-5241

ADVISORY GROUP ON ELECTRON DEVICES 1
SUITE 500
1745 JEFFERSON DAVIS HIGHWAY
ARLINGTON VA 22202

REPORT COLLECTION, CIC-14 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

AEDC LIBRARY 1
TECHNICAL REPORTS FILE
100 KINDEL DRIVE, SUITE C211
ARNOLD AFB TN 37389-3211

COMMANDER 1
USAISC
ASHC-IMD-L, BLDG 61801
FT HUACHUCA AZ 85613-5000

US DEPT OF TRANSPORTATION LIBRARY 1
FB10A, M-457, RM 930
800 INDEPENDENCE AVE, SW
WASH DC 22591

AIR WEATHER SERVICE TECHNICAL 1
LIBRARY (FL 4414)
859 BUCHANAN STREET
SCOTT AFB IL 62225-5118

AFIWC/MSO 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

SOFTWARE ENGINEERING INSTITUTE 1
CARNEGIE MELLON UNIVERSITY
4500 FIFTH AVENUE
PITTSBURGH PA 15213

NSA/CSS 1
K1
FT MEADE MD 20755-6000

DCMAO/WICHITA/GKEP 1
SUITE B-34
401 N MARKET STREET
WICHITA KS 67202-2095

PHILLIPS LABORATORY 1
PL/TL (LIBRARY)
5 WRIGHT STREET
HANSCOM AFB MA 01731-3004

THE MITRE CORPORATION 1
ATTN: E. LADURE
D460
202 BURLINGTON RD
BEDFORD MA 01732

DUSD(P)/DTSA/DUTD 2
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

DR JAMES ALLEN 1
COMPUTER SCIENCE DEPT/BLDG RM 732
UNIV OF ROCHESTER
WILSON BLVD
ROCHESTER NY 14627

DR YIGAL ARENS 1
USC-ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

DR RAY BAREISS 1
THE INST. FOR LEARNING SCIENCES
NORTHWESTERN UNIV
1890 MAPLE AVE
EVANSTON IL 60201

DR MARIE A. BIENKOWSKI 1
SRI INTERNATIONAL
333 RAVENSWOOD AVE/EK 337
MENLO PRK CA 94025

DR PIERO P. BONISSONE 1
GE CORPORATE RESEARCH & DEVELOPMENT
BLDG K1-RM 5C-32A
P. O. BOX 8
SCHENECTADY NY 12301

MR. DAVID BROWN 1
MITRE
EAGLE CENTER 3, SUITE 8
O'FALLON IL 62269

DR MARK BURSTEIN 1
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

DR GREGG COLLINS 1
INST FOR LEARNING SCIENCES
1890 MAPLE AVE
EVANSTON IL 60201

DR STEPHEN E. CROSS 1
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PA 15213

DR THOMAS CHEATHAM 1
HARVARD UNIVERSITY
DIV OF APPLIED SCIENCE
AIKEN, RM 104
CAMBRIDGE MA 02138

MS. LAURA DAVIS 1
CODE 5510
NAVY CTR FOR APPLIED RES IN AI
NAVAL RESEARCH LABORATORY
WASH DC 20375-5337

DR THOMAS L. DEAN 1
BROWN UNIVERSITY
DEPT OF COMPUTER SCIENCE
P.O. BOX 1910
PROVIDENCE RI 02912

DR WESLEY CHU 1
COMPUTER SCIENCE DEPT
UNIV OF CALIFORNIA
LOS ANGELES CA 90024

DR PAUL R. COHEN 1
UNIV OF MASSACHUSETTS
COINS DEPT
LEDERLE GRC
AMHERST MA 01003

DR JON DOYLE 1
LABORATORY FOR COMPUTER SCIENCE
MASS INSTITUTE OF TECHNOLOGY
545 TECHNOLOGY SQUARE
CAMBRIDGE MA 02139

DR. BRIAN DRABBLE 1
AI APPLICATIONS INSTITUTE
UNIV OF EDINBURGH/80 S. BRIDGE
EDINBURGH EH1 LHN
UNITED KINGDOM

MR. SCOTT FOUSE 1
TSX CORPORATION
4353 PARK TERRACE DRIVE
WESTLAKE VILLAGE CA 91361

MR. STU DRAPER 1
MITRE
EAGLE CENTER 3, SUITE 8
O'FALLON IL 62269

DR MARK FOX 1
DEPT OF INDUSTRIAL ENG
UNIV OF TORONTO
4 TADDLE CREEK ROAD
TORONTO, ONTARIO, CANADA

MR. GARY EDWARDS 1
TSX CORPORATION
2000 N 15TH ST, SUITE 1000
ARLINGTON, VA 22201

MR. RUSS FREW 1
GENERAL ELECTRIC
MOORESTOWN CORPORATE CENTER
BLDG ATK 145-2
MOORESTOWN NJ 08057

DR MICHAEL FEHLING 1
STANFORD UNIVERSITY
ENGINEERING ECO SYSTEMS
STANFORD CA 94305

RICK HAYES-ROTH 1
CIMFLEX-TEKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

DR JIM HENDLER 1
UNIV OF MARYLAND
DEPT OF COMPUTER SCIENCE
COLLEGE PARK MD 20742

MS. YOLANDA GIL 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

MR. MORTON A. HIRSCHBERG, DIRECTOR 1
US ARMY RESEARCH LABORATORY
ATTN; AMSRL-CI-CB
ABERDEEN PROVING GROUND MD
21005-5066

MR. MARK A. HOFFMAN 1
ISX CORPORATION
1165 NORTHCHASE PARKWAY
MARIETTA GA 30067

DR RON LARSEN 1
NAVAL CMD, CONTROL & OCEAN SUR CTR
RESEARCH, DEVELOP, TEST & EVAL DIV
CODE 444
SAN DIEGO CA 92152-5000

DR CRAIG KNOBLOCK 1
USC-ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

MR. RICHARD LOWE (AP-10) 1
SRA CORPORATION
2000 15TH STREET NORTH
ARLINGTON VA 22201

MR. TED C. KRAL 1
BBN SYSTEMS & TECHNOLOGIES
4015 HANCOCK STREET, SUITE 101
SAN DIEGO CA 92110

DR JOHN LOWRENCE 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

DR. ALAN MEYROWITZ 1
NAVAL RESEARCH LABORATORY/CODE 5510
4555 OVERLOOK AVE
WASH DC 20375

ALICE MULVEHILL 1
BBN
10 MOULTON STREET
CAMBRIDGE MA 02238

DR ROBERT MACGREGOR 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL REY CA 90292

DR DREW MCDERMOTT 1
YALE COMPUTER SCIENCE DEPT
P.O. BOX 2158, YALE STATION
51 PROPSPECT STREET
MEW HAVEN CT 06520

DR DOUGLAS SMITH 1
KESTREL INSTITUTE
3260 HILLVIEW AVE
PALO ALTO CA 94304

DR. AUSTIN TATE 1
AI APPLICATIONS INSTITUTE
UNIV OF EDINBURGH
80 SOUTH BRIDGE
EDINBURGH EH1 1HN - SCOTLAND

DIRECTOR 1
DARPA/ITO
3701 N. FAIRFAX DR., 7TH FL
ARLINGTON VA 22209-1714

DR STEPHEN F. SMITH 1
ROBOTICS INSTITUTE/CMU
SCHENLEY PRK
PITTSBURGH PA 15213

DR. ABRAHAM WAKSMAN 1
AFOSR/NM
110 DUNCAN AVE., SUITE B115
BOLLING AFB DC 20331-0001

DR JONATHAN P. STILLMAN 1
GENERAL ELECTRIC CRD
1 RIVER RD, RM K1-5C31A
P. O. BOX 8
SCHENECTADY NY 12345

DR EDWARD C.T. WALKER 1
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

DR BILL SWARTOUT 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

GIO WIEDERHOLD 1
STANFORD UNIVERSITY
DEPT OF COMPUTER SCIENCE
438 MARGARET JACKS HALL
STANFORD CA 94305-2140

DR KATIA SYCARA/THE ROBOTICS INST 1
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIV
DOHERTY HALL RM 3325
PITTSBURGH PA 15213

DR DAVID E. WILKINS 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

DR. PATRICK WINSTON 1
MASS INSTITUTE OF TECHNOLOGY
RM NE43-817
545 TECHNOLOGY SQUARE
CAMBRIDGE MA 02139

DR JOHN P. SCHILL 1
ARPA/ISO
3701 N FAIRFAX DRIVE
ARLINGTON VA 22203-1714

DR STEVE ROTH 1
CENTER FOR INTEGRATED MANUFACTURING
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIV
PITTSBURGH PA 15213-3890

DR YOAV SHOHAM 1
STANFORD UNIVERSITY
COMPUTER SCIENCE DEPT
STANFORD CA 94305

MR. MIKE ROUSE 1
AFSC
7800 HAMPTON RD
NORFOLK VA 23511-6097

DR LARRY BIRNBAUM 1
NORTHWESTERN UNIVERSITY
ILS
1890 MAPLE AVE
EVANSTON IL 60201

MR. LEE ERMAN 1
CIMFLEX TECKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

DR MATTHEW L. GINSBERG 5
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE OR 97403

MR IRA GOLDSTEIN 1
OPEN SW FOUNDATION RESEARCH INST
ONE CAMBRIDGE CENTER
CAMBRIDGE MA 02142

MR JEFF GROSSMAN, CO 1
NCCOSC RDTE DIV 44
5370 SILVERGATE AVE, ROOM 1405
SAN DIEGO CA 92152-5146

JAN GUNTHER 1
ASCENT TECHNOLOGY, INC.
64 SIDNEY ST, SUITE 380
CAMBRIDGE MA 02139

DR LYNETTE HIRSCHMAN 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

DR ADELE E. HOWE 1
COMPUTER SCIENCE DEPT
COLORADO STATE UNIVERSITY
FORT COLLINS CO 80523

DR LESLIE PACK KAELBLING 1
COMPUTER SCIENCE DEPT
BROWN UNIVERSITY
PROVIDENCE RI 02912

DR SUBBARAO KAMBHAMPATI 1
DEPT OF COMPUTER SCIENCE
ARIZONA STATE UNIVERSITY
TEMPE AZ 85287-5406

MR THOMAS E. KAZMIERCZAK 1
SRA CORPORATION
331 SALEM PLACE, SUITE 200
FAIRVIEW HEIGHTS IL 62208

DR CARLA GOMES 1
ROME LABORATORY/C3CA
525 BROOKS RD
ROME NY 13441-4505

DR MARK T. MAYBURY 1
ASSOCIATE DIRECTOR OF AI CENTER
ADVANCED INFO SYSTEMS TECH G041
MITRE CORP, BURLINGTON RD, MS K-329
BEDFORD MA 01730

MR DONALD P. MCKAY 1
PARAMAX/UNISYS
P O BOX 517
PAOLI PA 19301

DR KAREN MYERS 1
AI CENTER
SRI INTERNTIONAL
333 RAVENSWOOD
MENLO PARK CA 94025

DR MARTHA E POLLACK 1
DEPT OF COMPUTER SCIENCE
UNIVERSITY OF PITTSBURGH
PITTSBURGH PA 15260

DR RAJ REDDY 1
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PA 15213

DR EDWINA RISSLAND 1
DEPT OF COMPUTER & INFO SCIENCE
UNIVERSITY OF MASSACHUSETTS
AMHERST MA 01003

DR MANUELA VELOSO 1
CARNEGIE MELLON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
PITTSBURGH PA 15213-3891

DR DAN WELD 1
DEPT OF COMPUTER SCIENCE & ENG
MAIL STOP FR-35
UNIVERSITY OF WASHINGTON
SEATTLE WA 98195

MR JOE ROBERTS 1
ISX CORPORATION
4301 N FAIRFAX DRIVE, SUITE 301
ARLINGTON VA 22203

DR TOM GARVEY 1
ARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

MR JOHN N. ENTZMINGER, JR. 1
ARPA/DIRO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

DIRECTOR 1
ARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

OFFICE OF THE CHIEF OF NAVAL RSCH 1
ATTN: MR PAUL QUINN
CODE 311
800 N. QUINCY STREET
ARLINGTON VA 22217

DR OREN ETZIONI 1
DEPT OF COMPUTER SCIENCE
UNIVERSITY OF WASHINGTON
SEATTLE WA 98195

DR GEORGE FERGUSON 1
UNIVERSITY OF ROCHESTER
COMPUTER STUDIES BLDG, RM 732
WILSON BLVD
ROCHESTER NY 14627

DR STEVE HANKS 1
DEPT OF COMPUTER SCIENCE & ENG'G
UNIVERSITY OF WASHINGTON
SEATTLE WA 98195

MR DON MORROW 1
BBN SYSTEMS & TECHOLOGIES
101 MOONGLOW DR
BELLEVILLE IL 62221

DR CHRISTOPHER OWENS 1
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON ST
CAMBRIDGE MA 02138

DR ADNAN DARWICHE 1
INFORMATION & DECISION SCIENCES
ROCKWELL INT'L SCIENCE CENTER
1049 CAMINO DOS RIOS
THOUSAND OAKS CA 91360

DR JAIME CARBONNEL 1
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIVERSITY
DOHERTY HALL, ROOM 3325
PITTSBURGH PA 15213

DR NORMAN SADEH 1
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIVERSITY
DOHERTY HALL, ROOM 3315
PITTSBURGH PA 15213

DR JAMES CRAWFORD 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE OR 97403

DR TAIEB ZNATI 1
UNIVERSITY OF PITTSBURGH
DEPT OF COMPUTER SCIENCE
PITTSBURGH PA 15260

DR MARIE DEJARDINS 1
SRI INTERNATIONAL
333 RAVENSWOOD AVENUE
MENLO PARK CA 94025

ROBERT J. KRUCHTEN 1
HQ AMC/SCA
203 W LOSEY ST, SUITE 1016
SCOTT AFB IL 62225-5223

DR. DAVE GUNNING 1
DARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

MS. LEAH WONG 1
NCCOSC RDTE DIVISIION
53560 HULL STREET
SAN DIEGO CA 92152-5001

B. MCMURREY 1
NCCOSC RDT&E DIVISION
CODE 421
53560 HULL STREET
SAN DIEGO CA 95152-5001

GINNY ALBERT 1
LOGICON ITG
2100 WASHINGTON BLVD
ARLINGTON VA 22204

DAVID HESS 1
SAIC ATLANTIC PROGRAMS
ONE ENTERPRISE PARKWAY, SUITE 370
HAMPTON VA 23666

COL ROBERT PLEBANEK 1
DARPA/ISO
3701 N FAIRFAX DR
ARLINGTON VA 22203

ADAM PEASE 1
TECKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

JIM SHOOP 1
ISX CORPORATION
4353 PARK TERRACE DR
WESTLAKE VILLAGE CA 91361

DR ROBERT NECHES 1
DARPA/ISO
3701 N FAIRFAX DR
ARLINGTON VA 22203

DAVID SWANSON 1
NRAD
53118 GATCHELL RD
44207 BLDG 600 RM 422C
SAN DIEGO CA 92152

DR STEPHEN WESTFOLD 1
KESTREL INSTITUTE
3260 HILLVIEW AVE
PALO ALTO CA 94304

MAJ TOMMY LANCASTER 1
USTRANSCOM/TCJ5-SC
508 SCOTT DR
SCOTT AFB IL 62225-5357

JAMES APPLEGATE 1
MITRE
EAGLE CENTER 3, SUITE 8
O'FALLON IL 62269

SOFTWARE ENGINEERING INSTITUTE 1
CARNEGIE MELLON UNIVERSITY
4500 FIFTH AVE
PITTSBURGH PA 15213

DIRNSA 1
R509
9800 SAVAGE RD
FT MEADE MD 20755-6000

NSA/CSS 1
K1
FT MEADE MD 20755-6000

DCMAO/WICHITA/GKEP 1
SUITE B-34
401 N MARKET ST
WICHITA KS 67202-2095

PHILLIPS LABORATORY 1
PL/TL (LIBRARY)
5 WRIGHT STREET
HANSCOM AFB MA 01731-3004

THE MITRE CORPORATION 1
ATTN: E LADURE
D460
202 BURLINGTON RD
BEDFORD MA 01732

OUSD (P)/DTSA/DUTD 1
ATTN: PATRICK G. SULLIVAN, JR
400 ARMY NAVY DR
SUITE 300
ARLINGTON VA 22202

☆U.S. GOVERNMENT PRINTING OFFICE: 1997-509-127-47115

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.