AFRL-IF-RS-TR-2000-121
Final Technical Report
August 2000

JFACC CONTINUOUS PLANNING AND EXECUTION

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. B545

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

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

AFRL-IF-RS-TR-2000-121 has been reviewed and is approved for publication.

APPROVED:

JOHN SPINA
Project Engineer

FOR THE DIRECTOR:

NORTHRUP FOWLER, Technical Advisor
Information Technology Division
Information Directorate

JFACC CONTINUOUS PLANNING AND EXECUTION

Karen L. Myers

Contractor: SRI International
Contract Number: F30602-97-C-0067
Effective Date of Contract: 25 Apr 1997
Contract Expiration Date: 30 March 1999
Short Title of Work: JFACC Continuous Planning and Execution
Period of Work Covered: Apr 97 – Mar 99

Principal Investigator: Karen L. Myers
Phone: (650) 859-2004
AFRL Project Engineer: John Spina
Phone: (315) 330-4032

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by John Spina, AFRL/IFTD, 525 Brooks Road, Rome, NY.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	AUGUST 2000	Final Apr 97 - Mar 99

4. TITLE AND SUBTITLE
JFACC CONTINUOUS PLANNING AND EXECUTION

5. FUNDING NUMBERS
C - F30602-97-C-0067
PE - 63760E
PR - F062
TA - 01
WU - 01

6. AUTHOR(S)
Karen L. Myers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SRI International
333 Ravenswood Avenue
Menlo Park CA 94025-3493

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

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

Air Force Research Laboratory/IFTD
525 Brooks Road

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2000-121

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: John Spina/IFTD/(315) 330-3577

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The responsibilities of the JFACC (Joint Forces Air Component Commander) are made challenging by a number of factors. JFACC are generally complex, open-ended, and broad in scope. Operating environments are dynamic, unpredictable, and typically hostile. Uncertainty is inherent, with the result that complete and accurate knowledge of the world can never be attained. For these reasons, successful operation requires a mix of long-range planning, responsiveness to unexpected events, and dynamic adaption of activity. Constructed plans must be living objects that evolve and grow over time in response to new tasks, updated information about the world, and result from partial execution through portions of the plan. SRI's project within the JFACC program sought to address these problems by developing a prototype "Continuous Planning and Execution Framework" (CPEF) that provides automated tools for the development, maintenance, and execution of living plans for highly dynamic environments such as JFACC, using a range of selected artificial intelligence (AI) technologies.

14. SUBJECT TERMS
Planning, JFACC, Artificial Intelligence

15. NUMBER OF PAGES
92

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

List of Figures

1 Introduction

SRI International (SRI) is pleased to present this interim summary report to Rome Laboratory and the Defense Advanced Research Projects Agency (DARPA), for contract F30602-97-C-0067 entitled *JFACC Continuous Planning and Execution.* The project was originally scheduled for the three-year period of April 25, 1997 through April 24, 2000. This report describes work completed through December, 1998, at which time the initial JFACC program was terminated.

1.1 Project Overview

The responsibilities of the JFACC (Joint Forces Air Component Commander) are made challenging by a number of factors. JFACC tasks are generally complex, open-ended, and broad in scope. Operating environments are dynamic, unpredictable, and typically hostile. Uncertainty is inherent, with the result that complete and accurate knowledge of the world can never be attained. For these reasons, successful operation requires a mix of long-range planning, responsiveness to unexpected events, and dynamic adaptation of activity. Constructed plans must be *living* objects that evolve and grow over time in response to new tasks, updated information about the world, and results from partial execution through portions of the plan.

SRI's project within the JFACC program sought to address these problems by developing a prototype *Continuous Planning and Execution Framework (CPEF)* that provides automated tools for the development, maintenance, and execution of living plans for highly dynamic environments such as JFACC, using a range of selected artificial intelligence (AI) technologies. *Process management* lies at the core of CPEF (see Figure 1A), providing the ability to coordinate and control a broad range of activities related to the continuous evolution of living plans, including plan generation, execution tracking, situation monitoring, and plan adaptivity.

In some sense, CPEF can be viewed almost as a 'mini JFACC', providing a microcosm of the process management capabilities envisioned for the overall JFACC system, although for a narrow slice of the full problem. In particular, CPEF provides capabilities for workflow management, development and ongoing supervision of plan generation (as envisioned for the PlanGen subsystem), and specialized component technologies for plan generation, plan repair, and situation monitoring (see Figure 1B). The technology, although developed within the context of the JFACC domain, is generic and domain-independent, and could be readily transferred to other domains.

Automated technology is critical to managing the complexity of the JFACC domain. However, the extensive knowledge and experience required for effective operation means that automated tools will never completely replace high-level human decision making. For this reason, the automated tools within CPEF support substantial user involvement with overall process management. While CPEF can run in fully automated fashion, users can direct the operation of the system when desired, without having to immerse themselves in the low-level details of ongoing processes.

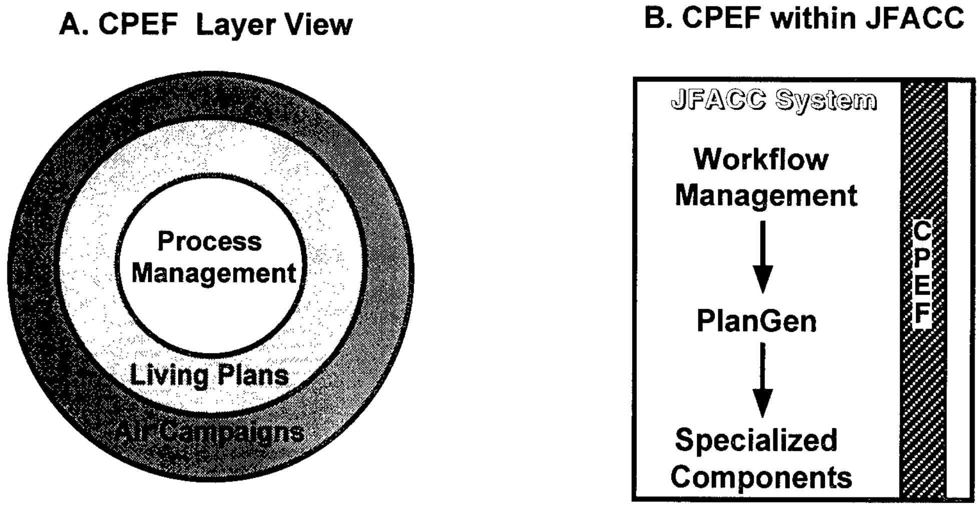

Figure 1: Two Perspectives on CPEF

CPEF leverages several sophisticated AI technologies as components. SIPE-2 [22] provides core planning services grounded in the *hierarchical task network* (HTN) model of plan generation [3]. These services support strategy-to-task refinement of objectives, and the ability both to identify and repair affected subplans in the face of changing world-state information and execution results. The Advisable Planner (AP) [16] supports user provision of advice to guide both plan generation and plan repair by the SIPE-2 planning system. AP thus enables users to direct planning tasks at high levels, letting the system manage the underlying details. The Procedural Reasoning System (PRS) [7, 14], a knowledge-based reactive control system that integrates goal-oriented and event-driven activity in a flexible hierarchical framework, is used both as a high-level reactive controller for the overall system, and to track execution through generated plans. Finally, CPEF builds on certain capabilities from the Multiagent Planning Architecture (MPA) [26] to provide distributed communication and plan storage services. Appendix A provides additional information about these foundational technologies.

CPEF is much more than a set of interfaces that simply enable these systems to interact. *Functional integration* of these technologies was required to enable tightly interleaved operation of these components. Furthermore, an explicit *process management* layer was added to control and coordinate the component technologies. Finally, the underlying technologies themselves had to evolve to support more sophisticated kinds of processing required for the JFACC tasks.

SRI's project was to concentrate on four main technical thrusts.

Flexible Process Management Provide intelligent management of planning and execution that is responsive to the dynamics of the operation environment.

Adaptive Behavior Provide situation monitoring, execution tracking, and plan repair to enable timely adjustment of plans and activities.

Robust Plans Develop plan generation methods that are sensitive to knowledge limitations and the dynamics of the operating environment, and so can reduce failures during plan execution.

User Guidance Support user directability of key aspects of overall system operation.

The work completed prior to the program termination date focused on the first two areas above. The third topic was to be a focus for the second year of the project: while substantial progress was made in the time available, the work was left unfinished. The fourth topic was scheduled for investigation in the final year of the project, although some advances had been made as of the writing of this report.

1.2 Summary of Accomplishments

The main accomplishments for the project are listed below, along with references to those sections of the report that describe them in detail.

- Prototype multiagent CPEF system, complete with extensive documentation and sample demonstrations (Section 2)
- Process management for plan generation, execution, monitoring, and repair (Section 3)
- Automated extraction of monitors from generated plans (Section 4)
- Execution monitoring for and recovery from a range of failure types (Section 5)
- Mixed-initiative plan generation and repair based on *user advisability* (Section 9)
- A customizable, instrumentable simulation environment for evaluating the effectiveness of monitoring, repair, and continuous planning capabilities (Section 6)
- Preliminary models for *open-ended planning* (Section 8)
- Task-level integration with the PlanGen framework and translation of plans into the JFACC Plan Representation (Section 7)
- Application of CPEF within the JFACC Cyberland domain (Appendix B)

1.3 Publications, Documentation, and Online Resources

The paper *Towards a Framework for Continuous Planning and Execution*, was presented by Dr. Karen Myers at the AAAI Fall Symposium on Distributed Continual Planning held in Orlando, Florida [18]. The paper summarizes the progress to date in realizing our vision for continuous planning and execution within the CPEF system.

The CPEF project web page, located at `http://www.ai.sri.com/~cpef/jfacc.html`, provides online documentation for the project, including instructions for loading and running CPEF, extensive descriptions of demonstrations, and sample Air Campaign plans produced and executed by CPEF.

1.4 Key Demonstrations

The following demonstrations were provided by SRI during the project. Extensive documentation for these demonstrations is available at the CPEF project web page, including operating instructions, detailed descriptions of the capabilities of the system, and sample plans.

- **Integrated Feasibility Demonstration (IFD 1.7)** *(December, 1998)*

  During IFD 1.7, a baseline CPEF system was demonstrated to show how SRI's planning and execution technologies could support the continuous development, monitoring and adaptation of Air Campaign plans. Given the short time frame in which this system was developed, the technical capabilities were somewhat shallow and brittle. Nevertheless, this baseline system successfully illustrated the value that AI planning and reactive control technologies can contribute to the automation of key JFACC capabilities.

- **Phase II Demonstration** *(June, 1998)* For the Phase II program demonstration, the capabilities of the baseline CPEF system were enriched in several ways. On the technical side, richer models of monitors, failure, and plan repair were implemented, and CPEF was migrated to a KQML-based distributed, multiagent platform. In addition, CPEF was linked to the PlanGen system so that it could respond to requests from PlanGen to create and repair options, and to store plans into the JFACC Plan Server. Finally, CPEF's knowledge bases were adapted to run in the JFACC Cyberland scenario.

- **Review by Col. McCorry at SRI** *(November, 1998)* For Col. McCorry's site visit to SRI, a detailed demonstration was provided of CPEF's capabilities to support incremental advice-directed plan generation, tracking of plan execution, and runtime plan repair in response to world-state changes and execution failures.

1.5 JFACC Program Involvement

SRI invested substantial effort in linking to other projects within the JFACC program. Most significant of these was the ongoing relationship with PlanGen, culminating in the integration of CPEF with the PlanGen system and the JFACC Plan Server for the Phase II demonstration (Section 7.1). SRI worked with researchers from University of Southern California/Information Sciences Institute (USC/ISI) and Carnegie Mellon University (CMU) to develop CommonP, a shared representation for the planning and scheduling technologies within the Technology Base (Section 7.3). In terms of integration, SRI built on USC/ISI's Adaptive Forms technology to develop an interface for specifying advice to be used in both the generation and repair of plans (Section 9.2). Links to additional JFACC projects are described in Section 7.

1.6 Report Overview

The remainder of the report consists of two main parts.

The first part of the report focuses on the results achieved within the project, covering Sections 2 through 7. Section 2 describes the architecture of the overall CPEF system, with detailed descriptions of key components provided in subsequent sections. Section 3 describes the *Plan Manager*, which serves as the central controller for the overall system. Sections 4 and 5 present our work on situation monitoring and plan adaptation. Section 6 describes the simulation framework that was developed to enable testing and evaluation of CPEF. Section 7 discusses SRI's involvement within the overall JFACC program, including actual and planned integration with other JFACC projects.

The second part of the report focuses on where the project was headed, including descriptions of work that was begun but not completed, and work that was proposed but not begun. Section 8 describes our preliminary work on *horizon-based planning* that was beginning to mature, while Section 9 discusses the limited work to date on *user guidance*. Section 10 presents key open challenges related to the project.

Section 11 presents our final conclusions. The appendices that follow provide descriptions of component technologies used within CPEF (Appendix A), summaries of key demonstrations (Appendix B), detailed instructions for loading and operating our final CPEF demonstration (Appendix C), and a glossary of acronyms and abbreviations (Appendix D).

2 CPEF Overview

Figure 2 provides an overview of the CPEF system. Boxes in the figure represent key functional capabilities, while arrows depict information flow. Attached to each box is the name of one or more technologies – the Advisable Planner (AP), the Procedural Reasoning System (PRS), SIPE-2 – that implement the associated functionality. Appendix A provides detailed descriptions of these underlying technologies.

CPEF employs an agent architecture that enables modular, distributed operation of these different functionalities (discussed further in Section 2.2). The agent-based design also facilities rapid integration with new technologies, which proved to be useful when integrating CPEF with other JFACC systems.

2.1 CPEF Components

The Plan Manager lies at the heart of the system, supervising plan generation, monitoring, repair, and plan execution. As such, it can be viewed as a kind of *workflow manager* for the task of building, maintaining, and executing living plans. The Plan Manager is always active, continuously monitoring the world for new tasks and information to which the system should respond. Section 3 describes the Plan Manager in further detail.

The Planner provides core plan generation and adaptation capabilities in accord with the strategy-to-task paradigm. Planning modes range from *fully automated*, to *interactive* and *advisable* planning. These planning capabilities are provided by a combination of SIPE-2 and the Advisable Planner, which is a module for advice processing and enforcement layered on top of SIPE-2. With advisable planning, users can express recommendations and preferences for the types of plans to be produced. Advisable planning is valuable both as a way of supporting user customizability of generated plans, and for enabling user-directed exploration of qualitatively different options.

The Plan Server provides a repository for storing multiple plans in an organized and principled fashion. While of limited use within the current system, we envisioned the Plan Server as an essential component for managing the large numbers of options and subplans that would be required to support long-term continuous planning.

Plan repairs are initiated by the Plan Manager in response to its monitoring of the situation and progress through plan execution. The Plan Repair module oversees adaptations to plans that are requested by the Plan Manager, communicating with the Planner and Plan Server as required.

The Simulator serves as a stand-in for the real-world execution of a plan. In particular, it accepts a given plan and computes simulated outcomes for the constituent actions in accord with customizable models of failures and delays. The outcomes are communicated to the Plan Manager to enable adaptation of strategies and plans based on partial execution outcomes.

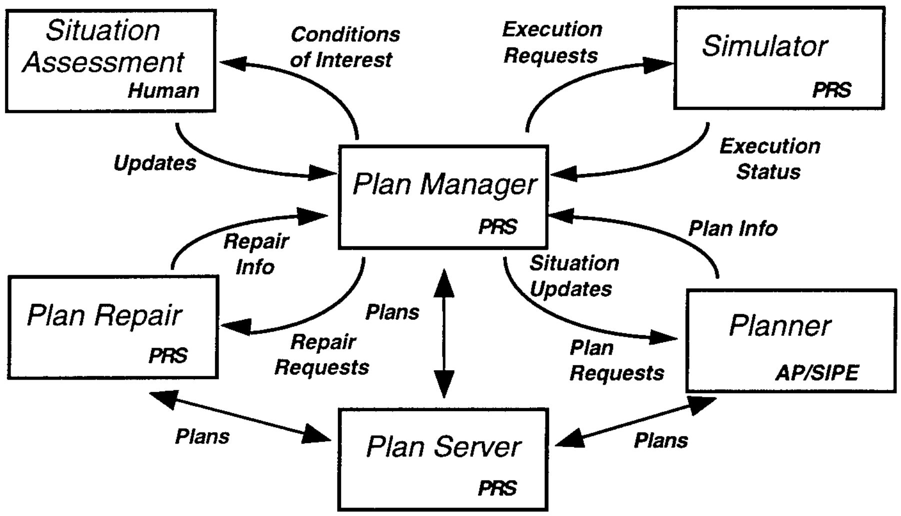

Figure 2: Functional Overview of CPEF

Simulated outcomes are modeled at the level of direct effects of basic actions within the plans (e.g., a mission completed successfully or unsuccessfully). It is the role of Situation Assessment (or Campaign Assessment) to provide 'roll-up' of action outcomes into higher-level evaluations of changes in the operating environment and progress through plan execution. Within CPEF, situation assessment is performed by a human providing asynchronous evaluations of a plan's execution. However, the framework is designed to accommodate automated streams of situation assessment information, which we anticipated would be provided by independent feeds from other technologies within the JFACC program.

Within CPEF, users can supply a range of information and requests to the system including assignment of tasks, situational updates, evaluation assessments, and advice for customizing plans. The system informs the user of critical events and activities, soliciting guidance when appropriate to direct problem-solving. Currently, users interact directly with the modules that they wish to influence or query, thus capitalizing on the interfaces developed for the technologies underlying each. In a completed system, an Interface agent would be desirable to provide a single point of entry for controlling and accessing the component systems.

2.2 CPEF Agent Architecture

Agent architectures provide many benefits that are essential to managing the complexities inherent to the JFACC problem. Agent-based designs encourage modular systems in which different capabilities can be encapsulated as distinct agents. These designs facilitate insertion of new technologies (i.e., *plug and play*), allowing for easy experimentation with different components. Agent designs support distributed operations, with different agents capable of running on geographically dispersed machines. In addition, agent frameworks support decentralized control in which individual agents can collaborate to produce results without a central controller having to manage every interaction.

The initial CPEF system (delivered in December, 1997) employed the agent-based framework of PRS. The PRS agent framework provides multiprocessing, asynchronous message-passing among agents, and clean separation of functionality among agents. However, PRS agents were designed to run as stand-alone entities, rather than serving as wrappers for other technologies. Thus, either a PRS agent must be created to provide 'wrapper' services for each component technology (a somewhat heavy-handed approach, since only the message-passing capabilities of PRS are required for CPEF), or the component technologies can only be accessed via blocking function calls (and hence are not true agents). Furthermore, PRS agents are restricted to running within a single executable, thus limiting distributivity.

Because of the limitations of the PRS agent model, a port of CPEF to the Multiagent Planning Architecture (MPA) [26] (see Appendix A.4) was completed in early 1998. MPA is an agent framework that provides a collection of services and capabilities designed to facilitate the management of complex, distributed plan generation tasks. These capabilities include the ability to run on distributed machines, interagent communication protocols, a dedicated Plan Server agent for storing plans and options, and wrappers for encapsulating arbitrary software modules as agents.

CPEF does not employ the entire MPA framework but rather incorporates two key MPA components into its infrastructure: the communication protocols and the Plan Server.

MPA Communication MPA provides a rich interagent communication framework, composed of a low-level *transport layer* and a higher-level *content layer*. The transport layer provides reliable point-to-point communication between agents, and is based on KQML message-passing [4]. The content layer consists of MPA-specific message formats and message-handling protocols that define a language for exchanging information and requests related to plans and planning activities.

MPA Plan Server The MPA Plan Server provides a central repository for plans and plan-related information, making this information accessible to all agents through a rich query language. The Plan Server is a *passive* agent in that it responds to messages sent by other agents but does not issue messages to other agents on its own initiative. The Plan Server accepts incoming information from agents, performs necessary processing, and stores relevant information in its internal representation. The Plan Server can be tasked to notify various agents of relevant planning events. The Plan Server supports different *views* of a plan, where a view constitutes some coherent subset of the content of a plan. For example, monitors, resource usage, and graphical representations constitute three distinct views of a plan. Views enable more efficient exchange of information: rather than having to retrieve an entire plan from the plan server to access certain information, agents can instead request a view limtied to the information that they require.

Within the MPA version of CPEF, interagent communication is performed using MPA message-passing rather than PRS's internal message facility. It should be noted that PRS is still a key element of CPEF, providing the technological foundations for the Plan Manager, Plan Repair, Plan Server, and Simulator agents.[1]

MPA was developed originally to provide general agent services for plan construction and evaluation tasks. For MPA to be used as the underlying agent infrastructure for CPEF, two key extensions were required. First, MPA protocols were defined to support the exchange of requests and results related to plan repair operations. Second, the conceptual framework was extended to include a class of *executor* agents along with corresponding communication protocols. Within CPEF, the PlanManager agent corresponds to an 'executor' agent of the latter type.

There were several tangible benefits from the use of agent technology within CPEF. The agent framework enabled rapid integration with Logicon's PlanGen system for the Phase II JFACC demonstration, through the creation of a *PlanGen Interface* agent that managed all interactions with the PlanGen system (described further in Section 7.1). The use of MPA within CPEF enables demonstrations that are easier to follow (due to the availability of multiple displays) and faster (due to the use of multiple distributed machines). The MPA Plan Server enables more organized and extensive control of the generation and use of multiple plans, as required for sophisticated forms of option generation, plan repair, and opportunistic planning.

[1]In certain situations, it is preferable to run with the PRS agent infrastructure rather than the MPA agent infrastructure (mostly for certain types of development tasks). For this reason, we have maintained both agent infrastructures within CPEF.

3 Plan Manager

3.1 Plan Manager Responsibilities

The *Plan Manager* is responsible for the overall control of system operation. Its main responsibilities are

- to control generation of plans and options for outstanding tasks
- to supervise execution of plans
- to provide knowledge management capabilities for plans and plan execution (e.g., monitor for key events, perform information-gathering tasks)
- to provide timely response to user requests and unexpected events
- to control adaptation of plans in response to plan failures

The Plan Manager can oversee multiple threads of activity at any given time. This multiprocessing enables, for example, one or more secondary plans to be activated in response to unexpected events in addition to the primary plan being executed (e.g., dispatch of a Search and Rescue mission to recover a downed pilot during execution of the main Air Campaign plan).

The term *Plan Manager* was selected for this module because its activities constitute a form of *process management* for the task of generating, maintaining, and executing a plan for the Air Campaign. The underlying technology, however, is general-purpose and can readily be tailored to a range of process management tasks. In particular, it could be applied to high-level workflow management for the overall JFACC process.

3.2 Direct vs. Indirect Execution

Previous work on reactive execution of plans focused on models in which an executor *directly* performs the activities specified in a plan. For example, the software controller for a mobile robot would initiate actual execution of actions by the robot (e.g., turning, increasing speed, or stopping) in the physical (or simulated) world.

This direct model of execution is inappropriate for the JFACC domain since the actions in Air Campaign plans must be performed by the pilots, intelligence operators and other operational personnel who are involved with the campaign. Instead, the JFACC operating environment requires an *indirect* model of execution, in which plan activities are performed by human agents in the real world rather than by a software controller. Within this indirect model, the role of an executor is to *track* the execution of the plan rather than to carry out the actions.

Tracking involves monitoring progress through the execution of the plan based on information (possibly incomplete) about the outcomes of individual actions within the plan.

While a seemingly subtle distinction, the difference between *direct* and *indirect* execution greatly impacts the design and behavior of an executor. With indirect execution, an executor may not have direct access to information about the success or failure of prescribed actions, or may not be able to determine whether those actions ever took place. Even when information is available, there may be a significant time lag between performance of an action and receipt of information about its status. Similarly, there may be substantial delays and cost in redirecting activities of the agents that are performing the actions in the plan.

The Plan Manager employs a *flow model* for tracking plan execution. This model involves waiting for reports on the outcome (*success*, *failure*, or *unknown*) of individual actions, in accordance with the temporal ordering relationships of actions in the plan. For example, if action A_1 precedes action A_2, the flow model dictates that the outcome of A_1 must be determined and appropriate responses taken before the outcome of A_2 can be considered. A more general approach would enable opportunistic tracking that can respond to action outcomes in arbitrary orders.

3.3 Plan Manager Implementation

The Plan Manager is built on top of the reactive control technologies of PRS. PRS provides an excellent technological match for the requirements of the Plan Manager because of its ability to combine event- and goal-driven processing in a uniform manner. This mixture is critical to supporting the monitoring and reactivity necessary to respond to key situational changes, along with the directed activities required to actively manage plans and respond to user requests.

The Plan Manager tracking mechanism was implemented as a variant of the standard hierarchical task execution mechanisms within PRS. In particular, specialized methods were defined for action achievement and condition testing that implement *tracking* rather than *execution* semantics.

3.4 CPEF Application Domain

CPEF was applied to the problem of continuously managing plan development and execution for Air Campaigns [10], with emphasis on achieving air superiority within a designated region. The plans in this domain are derived through hierarchical refinement of objectives for defensive and offensive air superiority, terminating at the level of the Air Tasking Order (ATO). The final plans, which require less than a minute to generate, contain several thousand nodes and can span as many as 25 refinement levels.

Extensive documentation for the demonstration system is available at the URL `http://www.ai.sri.com/~cpef/jfacc.html`, including operating instructions, detailed descriptions of the capabilities of the system, and sample plans. Here, we provide a brief overview of key concepts and contributions.

3.4.1 Air Campaign Planning Knowledge Base

The Air Campaign Planning Knowledge Base (ACP KB, or simply KB) encodes strategies (in the form of *operators*) to plan selected air objectives. Both offensive and defensive air superiority are supported, as is air supremacy. In addition to strategic information, the KB includes models for target networks and associated capabilities, coarse geographic information, aircraft capabilities, and threats.

In planning defensive air superiority, either a Protect or a Rollback and Protect strategy is adopted. Rollback involves the preemption of all threats along a hostile border. A Protect strategy means that, for each threatened center of gravity (COG), one of four protection alternatives is chosen: Combat Air Patrol (CAP) over the COG, barrier CAP (BARCAP) at the border, offensive counterair CAP (OCA CAP) at the origins of the threats, or preemption of the threats.

In planning offensive air superiority, a Breach and Extend strategy is used. A breach is a point at which the (Integrated Air Defense System) IADS is initially degraded enough to permit ingress by non-stealth aircraft. Once a breach is achieved, air superiority is extended over sectors to be attacked as part of the overall Air Campaign. Alternatives for achieving a breach are: one sector wide, two sectors wide, or all sectors along a hostile border.

Air superiority in a sector is achieved by reducing all surface-to-air missile (SAM) and intercept threats to that sector to acceptable levels. Such threats are modeled as networks whose components are either targets or more specialized networks. Components work in concert to provide a *capability* (e.g., SAM launch, intercept). Subsequent planning decisions are made as to which components of these networks are to be attacked, as well as how they are to be attacked. Some capability-specific strategies are encoded, such as blinding the IADS by attacking its ground-control and early-warning radars. Other strategies are general-purpose, in that they select some subset of network components to be attacked.

3.4.2 Demonstration Overview

Figure 3 provides a high-level overview of one run of the system. The timeline along the bottom shows exogenous events that lie outside of the system's control. Above that, activities are organized along three functional roles: the User, the Planner (encompassing plan generation and plan repair), and the Plan Manager. The system is designed so that the User plays an active role in the plan development and execution processes. The term Planner is used generically to refer to planning-related activities: plan generation, plan analysis, and plan repair. The executor is active at all times, providing three main threads of activity in parallel: *situation monitoring*, *execution tracking*, and *plan management*.

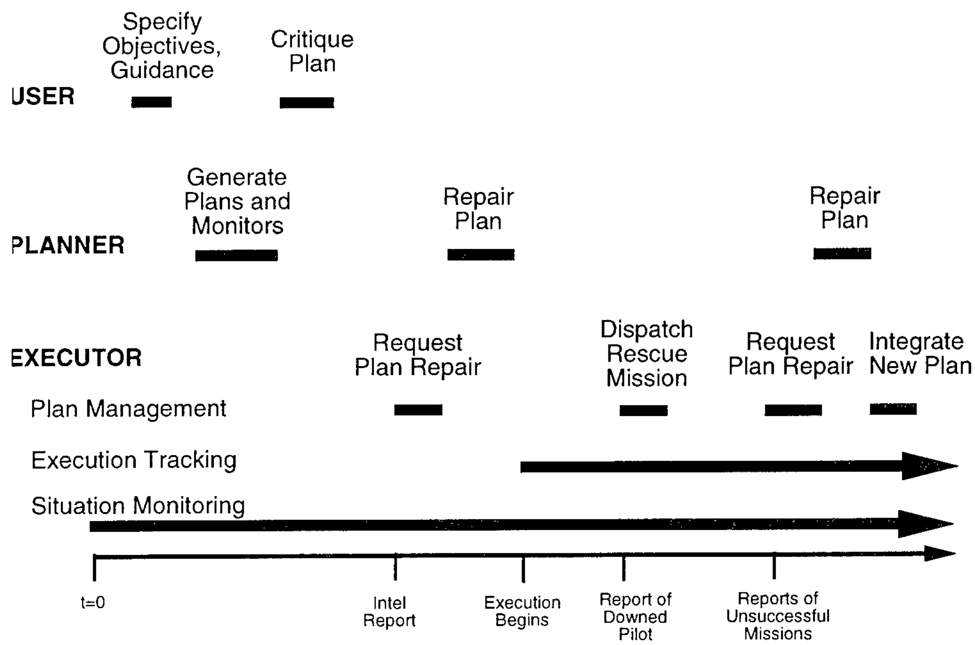

Figure 3: Demonstration Timeline

Activity begins in response to the User specifying air objectives for a given campaign, along with advice that reflects the commander's guidance for one or more courses of action to be developed. The Planner generates one or more plans to satisfy those objectives and advice, relative to its current knowledge of the operating environment. The completed plans are reviewed by the User, who can recommend changes to satisfy any outstanding concerns; the Planner then produces updated plans that incorporate the User's feedback. (Alternatively, the User could request a completely different plan through the specification of a new set of advice.) As planning proceeds, the Plan Manager monitors the environment for events that are relevant to the developing plan. Receipt of an intelligence update that invalidates parts of the plan will prompt a request that the Planner repair the affected portions of the plan.

Monitoring of the environment continues as execution of a selected plan commences. At some point during the execution, notification is received that a pilot has been downed; the system responds by instigating an appropriate activity (e.g., a Search and Rescue mission).

Later, an intelligence report is received indicating the possible presence of SA-11 SAMs in a critical sector. The JFACC decides to alter the current plan to attack C3 facilities that would support the SAMs to further degrade their effectiveness. CPEF performs dynamic repair of the current plan in order to incorporate missions in response to this decision.

In addition to monitoring for critical events of this type, the Plan Manager tracks progress through the execution of the plan to determine whether modifications are needed in response to the status of the execution. As an illustration, one generated CPEF plan contains a set of missions to neutralize a set of SAM sites, as a way of enabling access to a critical air sector. The receipt of reports indicating that more than a designated threshold of missions failed (i.e., an *aggregate failure*, as discussed further in Section 5) triggers plan repair to address the failure. By providing different advice to the system, users can influence the content of the revised plan. For example, advice could be provided to establish a second neutralization mission, or to modify the overall plan to eliminate the need for access to the air sector.

4 Monitors

The creation and deployment of *monitors* is a critical part of CPEF. We define a monitor to be an *event-response* rule for which detection of the specified *event* leads to execution of the designated *response*.

For CPEF, a plan consists both of a set of actions situated relative to a hierarchy of objectives, and conditions to monitor to ensure the likelihood of success for execution of the plan. In particular, monitors are explicitly created as a by-product of plan generation, as described further in Section 4.2. Monitors can also be created by a user (either interactively or programmatically), to track and respond to additional conditions of interest.

4.1 Monitor Taxonomy

Part of our efforts on monitoring involved developing a taxonomy of monitor classes designed to enable appropriate measured responses to critical detected events. The different classes derive from variations in both the types of events to be detected (for example, intelligence updates, changes in situation assessment information), and the nature of the responses (for example, generation of new options, localized plan repair, user alerts). We believe that this classification enables simpler and more modular specifications of monitors, for both automated and interactive approaches.

The main categories identified and explored to date were *failure*, *knowledge*, and *assumption* monitors.

Failure Monitors Failure monitors encode appropriate responses to failures that could occur during execution of a plan. One example of a failure monitor is to adapt a plan for establishing a breach of enemy IADS when a certain set of critical targets is not successfully eliminated.

Knowledge Monitors Knowledge monitors test for the availability of information about a world-state condition that is needed for decision-making. For example, it may be prudent to wait for an expected weather update before finalizing the choice of aircraft and munitions for a given set of missions.

Assumption Monitors The events in assumption monitors are changes in world-state information that violate assumptions upon which a given plan rests. For example, consider a plan for gaining defensive air superiority that was developed with a certain set of threats in mind. If additional threats arise, then new actions should be added to the plan to counter those threats. Similarly, if new information indicates that an expected threat no longer exists, then actions within the plan designed to counter that threat should be removed. Assumption monitors are particularly valuable in that they enable the detection of potential problems with a plan ahead of time, rather than having to wait for the problems to surface during plan execution.

4.2 Monitor Generation

CPEF supports user definition of a wide range of monitors. Additionally, certain kinds of monitors are generated automatically based on the content of a plan. In particular, we have developed and implemented a technique for the automated generation of assumption monitors from HTN plan derivation structures.

The algorithm for extracting assumption monitors involves a traversal of the HTN plan derivation structures generated by SIPE-2, collecting from each node those operator applicability conditions that are *dynamic* (e.g., troop movements, but not geographic conditions) and are not included in the effects of some preceding node in the plan (i.e., they must be satisfied in the initial world). The latter type of filtering eliminates large numbers of conditions that should not be monitored, since they are to be established by actions within the plan. Prespecified domain models identify a *response* to be performed when the applicability conditions are violated. Currently, there are three categories of responses: *alerts* for the user, *plan repairs*, and invocation of *standard operating procedures*. Additionally, the domain models indicate conditions for which assumption monitors are definable, thus filtering conditions whose violation is not significant. For example, weather conditions may fluctuate over time; their status can be disregarded until entry into a critical time window preceding key actions.

5 Failures and Repairs

Within CPEF, the Plan Manager determines when to initiate modifications to a plan. In contrast to many current systems, individual action failures are neither necessary nor sufficient for triggering plan repair. Rather, the Plan Manager supports a variety of models for interpreting failures and responding.

5.1 Generalized Failure Models

Within the AI community, models for detecting and recovering from plan execution failures have generally been limited to *precondition failures* and *action failures*. A *precondition failure* arises when associated preconditions for an action are not satisfied at the time the action is to be executed. For example, an action to dock at a port in an allied nation may have a precondition requiring transit approval for that port. An *action failure* results when the execution of an action does not attain its intended effects. For example, a neutralization mission may fail to hit its prescribed target.

These two types of failures, while important, cover only a small portion of the space of possible failures. In our initial CPEF system, we have taken a first step toward a more general framework that includes the following two types.

Unattributable Failures Failures are called *unattributable* if no individual action has failed or no assumption is violated, yet some assessment (human or automated) has deemed the current plan inadequate. For example, a commander may declare that a planned breach of enemy IADS has failed, despite the success of each constituent mission. Such a situation can arise either because the planning operators do not model the real world with sufficient fidelity, or simply because the commander has a conservative nature (e.g., he requires a high guarantee of neutralization before he is willing to fly subsequent missions through a sector). Unattributable failures need not be assessed by humans; evaluation functions could used to perform assessment of execution in a similar way.

Aggregate Failures In many situations, a single failure need not be cause for alarm. Indeed, good human planners often build redundancies into their plans to improve robustness. As a concrete illustration, Air Campaign plans often include extra missions above and beyond what is required to satisfy the objectives at hand to improve the likelihood of success.

Detection of unattributable and aggregate failures requires information beyond what is stored in standard plan dependency structures. Within CPEF currently, unattributable failures are identified by human assessors, while a simple theory of aggregate failures has been defined for Air Campaign plans.

5.2 Plan Repairs

For the JFACC domain (as for many others), it is important that plan repairs are *conservative* [19], in that they minimize changes to the original plan. Plans should evolve gradually, with small changes in the world or current goals resulting in proportionally small changes to the plan. Minimization of changes is important to ensure the continuity of the plan, and because of the potentially high costs of redirecting execution entities. To date, our focus has been on conservative repair based on analysis of plan dependency structures [21, 9]. In particular, we rely on the core methods defined previously within SIPE-2, along with several extensions (discussed below) that support more flexible forms of plan repair.

5.2.1 Arbitrary Plan Repair Points

Plan repair based on dependency structure analysis involves identifying a set of *root nodes* within the plan hierarchy that are the source of failures. Each such root has an associated *wedge* of lower-level tasks, which are removed from the plan. New subplans are then generated for each root node, if possible; otherwise, the process repeats for the parent of that node, terminating when the generation process succeeds.

To support the repair of unattributable failures, we have implemented a method that enables users to identify arbitrary root nodes whose wedges are to be replaced. Plan repair will then generate new subplans for the objectives on those roots.

5.2.2 Repair for Goal Generator Nodes

A second extension to the methods in SIPE-2 enables repair for *task generator* nodes. A task generator node differs from standard tasks within a plan in that it spawns a set of instances of a *task template* rather than a single task instance. The set of instances is determined by a special *creation condition*: an instance of the task template is created for each set of bindings that satisfies the creation condition.

Generator nodes provide a powerful representational capability that is critical for planning in many realistic domains. The operator knowledge base within CPEF relies extensively on generator nodes. For example, it contains an operator named *Protect-all-threatened-cogs* that can be applied to reduce threat levels to Blue COGs. That operator contains (among other subgoals) the goal generator

```
ACTION: GENERATE-GOALS
GOAL-GENERATOR: (defend-cog threatened-place when1 rating1)
CREATION-CONDITION: (blue-cog threatened-place)
```

To support plan repair, generated tasks whose creation conditions are violated are identified and removed from a plan. Furthermore, situation changes that result in the satisfaction of additional instances of the creation conditions lead to insertion of corresponding generated tasks into the plan.

5.2.3 Advisable Plan Repair

We extended the basic plan repair capabilities within SIPE-2 to incorporate user guidance (in the form of advice) as to what types of modifications should be made during plan repair.

5.2.4 Control

The Plan Manager supports *asynchronous* runtime repair: when problems arise during execution of a plan, the Plan Manager can invoke a module to produce a repaired plan while continuing to execute portions of the original plan that are unaffected by the problems. This mode of operation contrasts with *synchronous repair*, in which plan execution is halted while an alternative plan is generated. Asynchronous repair is critical in domains (e.g., military operations, robot control) where it is infeasible to halt all execution activities while repairing some portion of the overall plan.

5.2.5 Discussion

The plan repair capabilities within CPEF represent a start toward more flexible plan adaptation mechanisms. More work is required to produce the kind of general plan repair framework envisioned for the final CPEF system. Methods grounded in the analysis of dependency structures produce a plan that is proven correct with respect to the underlying domain model; here, correctness means that simulated execution of the plan will result in a world state where the original goals are satisfied. Even though this notion of correctness is somewhat weak (since unexpected events generally will occur, and the domain knowledge itself may be faulty), guarantees of correctness can be computationally expensive to secure. For this reason, a continuous planning system should ideally provide a spectrum of plan repair mechanisms ranging from the correct but costly minimal-perturbation, dependency structure methods to heuristic approaches that employ predefined domain-specific transformation rules (in the spirit of [2]), possibly trading correctness for efficiency.

6 Simulation Environment

The nature of the JFACC domain precludes evaluation of CPEF in an actual operational setting. Furthermore, no appropriate simulation environments in which to conduct experiments were identified by program management. For these reasons, we developed a PRS-based simulation environment called *SImulated FLexible EXecution* (SIMFLEX) to enable testing, evaluation, and demonstration of the continuous planning and execution capabilities of CPEF. SIMFLEX was intended to serve as an interim simulation environment until an appropriate high-fidelity simulator for the JFACC program was identified.

SIMFLEX provides a 'zero-fidelity' simulation capability that neither requires elaborate domain-specific action models, nor tracks state information in the simulated world. The only required inputs to SIMFLEX are the names of the possible actions that can be taken and conditions that can be tested, along with their associated argument lists.

For each possible action, SIMFLEX generates an Act for simulating the execution of the action.[2] These Acts are parameterized, thus enabling runtime-modifiable outcomes for the actions in the plan. To simulate a given plan, SIMFLEX traverses through its nodes, invoking the generated Acts using the standard task refinement methods within PRS. By building upon the infrastructure of PRS in this manner, development of the simulation environment required relatively little effort.

Users can provide either fixed global defaults or probability distributions that determine the rates at which actions or condition tests succeed, fail, or yield no information about their execution, as well as the duration of actions. Additionally, users can specify overriding success and duration rates for individual actions, conditions, and tasks, or classes of such objects.

As noted above, initializing SIMFLEX requires information about the possible actions and conditions within the domain. Our expectations were that the CPEF knowledge bases and planning operators would be expanded during the project to increase both their coverage and sophistication. For this reason, we developed tools that automate the extraction of the required domain information from the CPEF knowledge bases, thus enabling automatic initialization of SIMFLEX environments.

CPEF collects a range of statistics related to plan execution, including execution times, types of failures encountered, and repairs initiated. These statistics were to form the basis for quantitative evaluation of the effectiveness of the continuous planning and repair mechanisms being developed within CPEF.

The interactions between the Plan Manager and SIMFLEX were designed so as to make transition to a higher-fidelity simulation environment straightforward. Messages from the Plan Manager to SIMFLEX consist of requests to initiate or abandon the simulation of a designated plan. Messages from SIMFLEX to the Plan Manager consist of status reports on the execution of actions. Communication between the Plan Manager and SIMFLEX is 'handshaked': after issuing a status report for a particular action or test condition, SIMFLEX

[2] Acts are the basic unit of action representation within CPEF.

waits for a confirmation from the Plan Manager before proceeding. This synchronization process provides a simple mechanism to prevent the simulation from getting too far ahead of the Plan Manager. In a more realistic and flexible simulation environment, such handshaking would not be required.

In summary, SIMFLEX provides a flexible, tailorable environment in which to perform extensive evaluation of continuous planning and execution capabilities.

7 Program Interactions

This section summarizes our participation within the overall JFACC program. In particular, it covers our relationship and integration with PlanGen and the JFACC Plan Server, our involvement with the CommonP effort, and our collaborations with other JFACC projects.

7.1 PlanGen Integration

We completed two TIEs (Technology Integration Experiments) to enable the integration of CPEF into the PlanGen (and hence, JFACC) system. The first TIE focused on the development of a translator from SRI's Act language (used to represent plans and processes within CPEF's component technologies) and the JFACC Plan Representation (also known as the JFACC C2 Schema). The second TIE involved a task-level integration of CPEF with the PlanGen framework using the JFACC program *event channels*.

7.1.1 Act to C2 Schema Translator

The Act to C2 Schema translator maps a multilevel plan from SRI's Act representation to the C2 Schema, as defined by the JFACC Object Model (JOM) Version 2.0 (released April 21, 1998).

While the Act and C2 Schema representations both embed hierarchical models of plans, they differ in the specifics of their approaches to structuring this information. The Act formalism adopts a *level-oriented* organization in which a hierarchical plan is represented as a collection of complete plans at multiple levels of refinement (each represented by a so-called *action network*) with links between a node in a given level and its descendants in the subsequent level. The C2 Schema employs a *node-oriented* organization, where a given node is connected to its descendants but connections to other nodes within that action level are left implicit. To translate to the C2 Schema, each *action* (i.e., *objective*, *task*, or *activity*) in a given level, along its associated conditions and effects, is translated to a *JfaccAction* object with one option. The collection of actions for that option is the set of nodes in the next level of the *action network* whose parent is the current action.

One major problem with the JOM defined for the Phase 2 demonstration is that elements of the JOM object class used for action arguments, namely *JfaccMmDirectObject*, are represented as strings. Similarly, elements of the object class *JfaccConstraints*, which are used to define action preconditions and intended effects, are also represented as strings. As such, the translation of CPEF-generated plans into the C2 Schema can be used for little other than display purposes. For other systems within JFACC to make use of plans stored in the Plan Server, structured representations of actions and constraints must be incorporated into the JOM.

During development of the translator, several additional shortcomings of the JOM became apparent. For example, coverage of the JOM class *JfaccActionVerbs* that denote the possible actions within a plan was inadequate for the scope of the plans that CPEF generated. Additionally, ordering constraints among actions within a plan were not recorded.

The translator covers the main structure of an Act plan, including actions with their preconditions and effects, hierarchical refinement links, and precedence relationships among actions. However, it does not encode the entire content of an Act plan. Constructs that are omitted include certain rationale information (such as the choice of operators used to refine objectives, or advice that influenced planning decisions), process information (alternative strategies that could be applied for individual objectives), and complete accounts of the dependency structures among plan nodes. The Phase II JOM did not readily support the encoding of this type of information.

Translation of the plans from Act to the C2 Schema is efficient; however, installation of the plans into the Plan Server proved to be a bottleneck. While the plans generated by CPEF for the Phase 2 demonstration are of only moderate size (containing approximately 2200 nodes within the Act representation), their C2 Schema translations required close to 15 minutes each to store into the Plan Server. For this reason, the CPEF plans had to be preloaded into the Plan Server for the Phase 2 demonstration. Clearly, a more efficient design would be necessary to support real-time installation of plans into the Server.

7.1.2 Task-level Integration of CPEF and PlanGen via Event Channels

Task-level integration of the CPEF and PlanGen systems was achieved through the JFACC *event channel* mechanism. The integration supports the processing of and response to requests from PlanGen to CPEF for the generation of Courses of Action (COAs), and plan repairs in response to stated situation changes. For each type of request, CPEF returns a single option. Extensions to support additional requests/responses would have been straightforward to implement (and were planned for Phase 3).

A dedicated agent (called *PG Interface*) was added to the CPEF framework to manage interactions with all agents defined in the PlanGen system (see Figure 4).[3] To support the requirements of the Phase 2 Demonstration, CPEF interacts with two PlanGen agents: *COAFormulationAgent* and *PlanRepairAgent*. This integration of the CPEF and PlanGen systems employs a pair of event channels (one for each direction) for each category of event passed between CPEF and PlanGen.[4]

[3] While the delivered version of CPEF supports the *PG Interface* agent described here, the CPEF demonstrations at IFD 2.0 did not make use of this agent, due to the impracticality of performing real-time installation of plans into the JFACC Plan Server (as described in the previous section).

[4] The use of so many specialized event channels was mandated by the requirements of the PlanGen system. We recommended a simpler design in which there is a single channel for each direction of communication between the agents, with different categories of events supported within a single channel. The current approach will be problematic when integrating large numbers of agents, since there will an extremely large number of channels to maintain. Furthermore, extending the system to support additional events will be unduly burdensome, requiring creation of $2n$ new channels when n agents need to monitor the new event class.

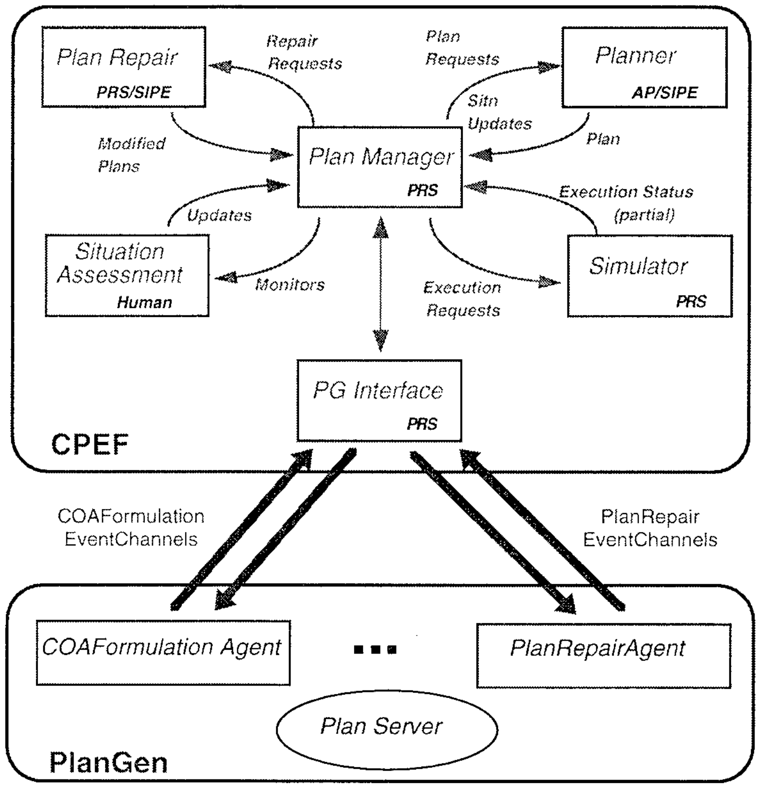

Figure 4: CPEF/PlanGen Integration

The *PG Interface* agent subscribes to and monitors the relevant Event Channels, takes actions to respond to appropriate events, and dispatches events to PlanGen as appropriate. Within the current integration, all interactions are initiated by PlanGen. However, more general forms of control are readily supportable. The *PG Interface* agent services requests from PlanGen by issuing requests to CPEF's Plan Manager agent, and responding to the results it returns as appropriate. Encapsulation of this interface into a distinct agent has numerous benefits, including distributability and modularity.

As with the Act to C2 Schema translator, the task-level integration of the CPEF and PlanGen systems was impaired by a lack of structured representations for events. Because elements of the class *JfaccEvents* used to represent events within the current JOM are represented as strings, the integration required the specification of hard-coded tables that map complete event strings to appropriate CPEF representations of events.

7.1.3 Discussion

Progress on these two TIEs was hampered by two main problems. First, explicit specifications for the C2 Schema and the JFACC Events were not finalized until just before the start of IFD

2.0. Second, the full TIF could not be run at remote sites until the weekend preceding the IFD. As a result, significant debugging of our translator and event-processing code could not be undertaken until our arrival in San Pedro for the IFD. Nevertheless, we were able to complete the integration in time for demonstration at the IFD.

The JFACC Plan Representation (JPR), as formulated for the Phase II demonstration, had several problems. First and foremost was the lack of a structured representation for terms, constraints, and activities. In particular, strings were used to represent complex objects. As a result, much of the content of a plan that was translated into the JPR could be interpreted only by humans, not by automated systems. This precluded, for example, the ability to pass a subplan produced by CPEF and translated into JPR to INSPECT, since INSPECT would have no way to interpret the content of the JPR strings. Another problem with the JPR was that the *node orientation* makes it difficult to understand the context of planning decisions.

The JFACC Events language that was to be used to coordinate operations of different modules relative to the Plan Server also lacked sufficient detail to be usable. Events were simply predefined strings, making it difficult to develop a framework for the exchange of complex events. Furthermore, the event channel mechanism, which required two event channels for each event type, would never have scaled to the level required for the JFACC system.

7.2 Transition to the Cyberland Domain

The COA generation capabilities within CPEF employ a collection of Knowledge Bases (KBs) that inform the process of refining high-level objectives. The primary KBs within the system are *operators* that encode strategic information for objectives refinement, *target information* including target catcodes, BE (basic equipment) numbers, geolocs for individual targets, and aggregations of targets into networks, *geographic data*, *asset classes*, *resource information*, and *intelligence information* (i.e., COG analyses, threat assessments),

The baseline CPEF system from December, 1997, demonstrated Air Campaign Planning in an extended version of the Granola scenario from the ARPI TIE-97-1 system, which focused on a conflict among fictional geopolitical entities in the western United States and Mexico. Transition to the Cyberland scenario required changes to almost all of the CPEF KBs.

Performing the requisite modifications to the Cyberland KBs was straightforward once the appropriate scenario information was provided by the JFACC Demo Team. Development of the new KBs required roughly 4 to 5 man-days of effort. Much of that time was spent on building models for aggregating targets into appropriate target networks, based on the functionality that those networks can provide (e.g., IADS networks, communication networks, energy distribution).

It is worth noting that two of the CPEF KBs required no modifications to support the Cyberland scenario. As one might expect, the *asset classes* KB, which documents the capabilities of various types of munitions, force structures, equipment, and aircrafts carried over directly. More surprising was the fact that the Operator KB did not require any modifications.

Although the strategies encoded by the operators had been developed for Air Campaigns to be waged over a large land mass, they were directly reusable within the small-island setting of the Cyberland scenario.

7.3 The CommonP Language

In an attempt to address some of the problems that were identified with the JPR, particularly in regard to its suitability for enabling interoperability among automated planning and scheduling technologies, several members of the Technology Base (TechBase) teamed to define a shared representation that would support interaction between TechBase technologies.[5] The motivation for this effort was not to replace the JPR. Rather, it was hoped that the effort would quickly enable exchange of semantically meaningful information among TechBase systems. The expectation was that the results would lead to a unified set of recommendations from the TechBase for influencing future revisions of the JPR so that it would more readily accommodate the representational requirements of the automated technologies.

SRI participated in the kickoff meeting for this effort, held at USC/ISI in the fall of 1997. That meeting led to an initial outline for the language, christened CommonP, which unified the basic representations from INSPECT, Loom, SIPE-2, and PRS. Since that time, the language has been extended and refined, particularly to include scheduling constructs based on CMU's work.

Subsequent to the kickoff meeting, SRI developed a set of planning operators and plans that served as test cases for the CommonP design. The operators and plans are based on models from the Air Campaign Planning domain that was used for the baseline CPEF demonstration in December, 1997. An initial encoding of these test cases within CommonP was performed as a means of testing its representational adequacy and usability.

7.4 Advice Interface Based on Adaptive Forms

As described in Section 9.2, SRI developed an Advice Interface for inputting user guidance to the plan generation and plan repair processes, based on the Adaptive Forms Interface from USC/ISI.

[5] The members of this group were Stephen Smith from CMU, Karen Myers from SRI, and Yolanda Gil, Robert MacGregor, and Bill Swartout, from USC/ISI.

8 Open-Ended Planning

The development of methods to support *open-ended* planning was to be a key research focus during Phase III of the project. Our investigation of open-ended planning began in September of 1998, but was cut short by the early termination of the program. Here, we summarize the approach that we were pursuing,

8.1 Overview

Traditional plan generation methods (including those within CPEF) are designed to create end-to-end solutions. In particular, these technologies assume that there is a readily identifiable 'end state', and that plans from the initial state through that end state both can and should be generated. For domains such as JFACC, such assumptions are invalid because no clear end state can be defined at the outset of a conflict. Furthermore, uncertainty about the operating environment and the expected effects of planned actions precludes generation of plans that extend far into the future. Rather, domains such as JFACC require the creation of *open-ended* plans that grow and evolve in response to the dynamics of the environment.

Two key problems must be addressed to support such open-ended planning, namely, *recognition of planning horizons* and *incremental planning*.

Planning Horizons Planning horizons constitute boundaries on what makes sense to plan given the current knowledge about the world state and the current goals. Horizons can be both *temporal* (i.e., limits on how far into the future to plan) and *abstraction-related* (i.e., limits on the level of refinement to which to plan). Effective planners, whether human or automated, must take horizons into account when generating plans to avoid constraining activities unduly, or making poor choices because of missing information.

Incremental Planning Incremental planning techniques are required to enable the growth and evolution of open-ended plans in response to situation changes or partial execution results. To date, we have explored both *passive* and *active* methods for operating with respect to horizons. The passive methods rely on monitors to detect when relevant information has been received that could change planning horizons, thus enabling incremental updates to a plan. The active methods initiate activities to gather relevant information that will enable planning to proceed.

Our initial focus was on identifying and accommodating *abstraction-related planning horizons*, which result from the unavailability of information required to enable expansion of objectives. *Temporal planning horizons*, which relate to how far out into the future to plan, were to be addressed later in the project.

8.2 Planning Relative to Abstraction-related Horizons

Our general approach to abstraction-related planning horizons involves categorizing information used in planning according to the type of response that should be undertaken when that information is missing or uncertain. We identified several different categories of information whose unavailability or uncertainty could impede plan generation. For these categories, we defined corresponding strategies for responding when this information is needed during plan generation but not available.

Case Analysis and Contingency Planning For situations where there is uncertainty but the number of alternatives is small, case analysis can be performed to generate options for each. For example, when planning to defend a Blue carrier from attack, it may be that threats could originate from only a small number of enemy bases. In such a case, it would be tractable to generate contingency plans for each possible threat source [20].

Goal Postponement for Expected Information Certain types of information may be expected to become available as the planning process unfolds. Some of this information may be provided one time only (e.g., the commander's Rules of Engagement near the outset of the planning process) or may arrive on a periodic basis (e.g., weather reports). We say that an objective that cannot be planned without knowledge of these conditions has an *information dependency* on that knowledge.

Other types of information may be generated as a by-product of the planning process itself. For example, the expected availability of resources for attacking enemy IADS for offensive air superiority objectives will depend on how many resources have been allocated for establishing defensive air superiority earlier in the plan. We call this type of connection a *goal dependency*: in this case, expansion of the particular goal depends on information that is a consequence of expanding some other goal.

When it is expected that certain information will be forthcoming, an appropriate response may simply be to delay refinement of a goal until that information arrives. We call this strategy *goal postponement*. Accompanying monitors should be created that will trigger reconsideration of postponed goals once the required information becomes available. Time-out conditions will also be required to ensure that the planner does not wait indefinitely for information that never arrives. In particular, in the case of such time-outs, either assumptions will have to be made or contingency planning undertaken to provide relevant options given the current state of knowledge.

Knowledge Gathering For certain types of missing information, there may be activities that can be undertaken to acquire the information when it is needed for planning. For example, if intelligence about enemy threats within a particular sector is required, an ISR mission could be initiated to obtain the missing information. Upon receipt of the requested information, plan generation could proceed as usual.

Assumptive Planning Because of the inherent uncertainty in the JFACC operating environment, plan generation methods must accommodate uncertainty. When required information is not available and the number of possibilities is too large to permit generation of contingency plans for each, assumptions must be made as to the most likely and most critical eventualities, with plans generated for each. As with goal postponement, monitors should be created to identify (if possible) when incorrect assumptions have been made, as additional information becomes available. When incorrect assumptions are detected, elements of the plan that rely on those assumptions should be reconsidered.

The generative planner module within CPEF (namely, SIPE-2) already supported limited forms of case analysis and contingency planning, in terms of generating different subplans to handle a small, fixed set of eventualities. To complement these capabilities, we began implementation of methods for assumptive planning and goal postponement within CPEF. Knowledge-gathering methods were to be implemented later, in conjunction with the determination of appropriate modules for collecting JFACC-related knowledge.

Our implementation of assumptive planning involves introducing a set of deductive rules that capture default or assumption information. If that information is missing, these rules can be triggered to enable deduction of likely assumptions. In the future, we intended to support interactive specification of assumptions: the user would be presented with the set of possible cases and asked to identify those for which alternatives should be generated. We also intended to link generated assumptions to the CPEF monitoring facility so that when assumptions are violated, dependent planning decisions could be reconsidered.

The implementation of goal postponement involved generalizing the control module within SIPE-2 to delay planning for goals when required information was not yet available. Determination of when to postpone planning of a goal is based on properties stored about individual predicates. In our implementation, SIPE-2 can postpone affected planning decisions when information is missing, but the link to CPEF's monitoring facility was not completed.

9 User Guidance

User guidance was to have been a focus of SRI's JFACC project in Year 3. However, we devoted some effort during the 20 months of the project to leverage advances from SRI's Advisable Planner (made within the ARPI program) that were of relevance to user guidance for CPEF. In particular, the following capabilities were added to CPEF: (1) an *incremental advice change* mechanism, and (2) an Advice Editor built on USC/ISI's Adaptive Forms system. We provide brief summaries of those efforts here.

9.1 Incremental Advice Change

SRI introduced an incremental *advice change* capability into its Advisable Planner, in work sponsored by the ARPI program. After creating a plan for some initial set of advice, the user can employ this mechanism to add or remove advice and request that those changes be reflected in the plan. The Advisable Planner will then perform a minimal set of modifications to the plan to enact those changes, leaving unaffected portions of the plan intact.

Previously within the Advisable Planner, users could change advice but the entire plan would then have to be regenerated. With such whole-scale regeneration, no guarantees could be made that aspects of the plan that met with user approval would be preserved. The new advice change capability enables user to explore the space of plans incrementally, by advising the system on specific aspects of the current plan that should be modified. Such a capability is essential for effective mixed-initiative plan development frameworks.

CPEF incorporates the incremental advice change mechanism, thus enabling users to develop plans in an incremental, exploratory fashion. In addition, we extended the advice change mechanism to enable its use during plan repair. As such, users can easily adapt executing plans through a change in guidance to produce a plan that better reflects current user preferences and insights gained during plan execution.

9.2 Advice Interface Based on Adaptive Forms

In ARPI-sponsored work, an Advice Editor for the Advisable Planner was built on top of USC/ISI's Adaptive Forms technology. In contrast to the domain-independent form-based editors available previously for advice specification, this new editor enables advice to be specified using a restricted subset of English.[6] As such, it provides a more natural and accessible interface for providing guidance to the underlying planning technology.

The Adaptive Forms advice editor was developed to support specification of advice to the ARPI TIE-97-1 system demonstrated at EFX'98. As such, it employs a grammar of legal advice expressions that, while grounded in a general Air Campaign Planning knowledge base,

[6] A natural language advice interface was built previously by SRI; however, that interface is tailored to a travel-planning domain.

are tied to the Granola scenario employed within the TIE-97-1 system. We ported the editor to provide an advice interface for CPEF, thus improving the usability of the advice-giving mechanism. This port involved adapting the geography to the Cyberland domain and extending the grammars to support a broader range of advice.

The use of the Adaptive Forms editor within CPEF was also motivated by the desire to promote further integration between CPEF and other USC/ISI technologies that make use of the Adaptive Forms framework, specifically INSPECT and the Strategy Development Assistant (SDA). In particular, we had envisioned that at some later point in the program, an Adaptive Forms interface might provide a single point of entry for users to provide a range of information (including objectives, advice, and measures of merit) to our systems.

10 Open Challenges

CPEF, while unfinished, already provides many of the foundational capabilities required for continuous planning and execution in highly dynamic and complex worlds. These capabilities include an agent-based architecture, rich monitoring and repair strategies, flexible integration of plan generation and execution, and highly reactive and adaptive problem-solving capabilities. More is required, however, to produce a truly continuous planning and execution system.

In this section, we summarize the technical directions that we had intended to pursue, and the relevance of that work to the overall JFACC objectives for continuous planning and execution technology.

10.1 Advanced Process Management

The current CPEF provides base-level process management for continuous planning and execution. However, advances are necessary to provide truly effective and robust behavior. Two areas that we intended to pursue are *extended control strategies* and *complex monitors*.

10.1.1 Extended Control Strategies

CPEF requires richer control strategies for responding to unexpected events and failures. Here, we describe three areas for future work.

Continuous Plan Repair CPEF currently can respond to individual failures, triggering plan repair processes that will fix the problem and merge the repaired plan with ongoing activities. However, *cascaded faults* where additional faults occur while a single fault is being addressed are not adequately handled. In particular, the system will postpone addressing any subsequent failures until the initial problem has been fixed. Ideally, the system should be able to interrupt ongoing repair processes to incorporate additional failure information so that problems are addressed as they arise. The result would be a system in which planning and execution are much more tightly intertwined.

Heuristic Plan Repair Current repair strategies within CPEF are limited to correctness-preserving methods grounded in analysis of dependency structures (as discussed in Section 5). A continuous planning system should ideally provide a spectrum of plan repair mechanisms ranging from the correct but costly minimal-perturbation, dependency structure methods to heuristic approaches that employ precompiled strategies for directly repairing plans that trade correctness for efficiency. Control models for deciding when to employ which strategy are necessary. When heuristic methods are employed, verification methods should be run in the background to identify and repair possible problems in the heuristic solutions.

Plan Transition Models Work on plan repair has mostly ignored issues related to *switching* between different plans. Transition is one consideration: taking the steps necessary to terminate activities from the current process and redirect to the new process. Cost is a second consideration: repair strategies need to incorporate realistic models of the expense involved in redirecting ongoing activities. For example, a commander may recognize that an in-progress plan is no longer the best option, but may continue with it because assets have already been deployed in accordance with its underlying strategy.

10.1.2 Complex Monitors

Our work on generalized models of failures presents a start toward a more flexible and powerful framework for execution monitoring. However, additional work is necessary to provide the expressive monitoring capabilities required for the complexities of the JFACC domain.

Just as the strategy-to-task objectives hierarchy is critical to understanding *why* a particular mission is included in an ATO, it is also essential for situating and interpreting failures. For example, the failure of a single mission to eliminate an assigned target generally is not a problem, since good planners build redundancy into their plans to accommodate such isolated failures. However, the failure of several missions related to the same objective of neutralizing a critical enemy IADS network probably would be significant. Current models for execution monitoring do not readily support this type of contextualization.

We had intended to develop a framework to support *complex monitors* that could detect the occurrence of sets of events that stand in some interesting semantic relation to each other. These relations could be defined temporally (e.g., Overflights followed by significant troop movement), spatially (e.g., Multiple successful SAM launches within Sentani sector), or in terms of objectives (e.g., Failure of more than two air missions related to neutralization of a key IADS installation). This framework was to have been layered on top of the monitoring facility currently available within CPEF.

10.2 User Guidance

Automated technology is essential to managing the complexity and dynamicism of the JFACC operating environment. Full automation, however, is neither desirable nor feasible. As such, it is essential for the JFACC program to develop mechanisms by which users can effectively interact with and control the new technologies that are developed.

For Year 3, SRI had proposed to build an advice-taking interface for user control of the planning and execution processes that would extend and complement ongoing work by SRI on the *Advisable Planner* project within the ARPI program. In particular, advice would serve as a kind of high-level control language for plan execution, replanning, and repairs, couched in terms that are natural and meaningful for the users. While the Advisable Planners project emphasized *product-related* advice that describes characteristics of the desired plan, here we were to focus on *process-related* advice that supports recommendations for both the plan management process itself (*management advice*) and how to carry out plans (*execution advice*).

For example, a commander who is developing an Air Campaign plan within the CPEF framework might supply the following management advice to control the level of detail to which the planner operates:

Plan the sorties in Sentani Sector down to the equipment level to prevent overallocation of resources.

This would result in a rough plan that might help the commander evaluate the viability of a particular strategy. Alternatively, the commander may provide the following management advice to explore a range of possibilities:

Show me three options for breaching enemy IADS that trade risk against resource requirements.

The following management advice would instruct the system on how to proceed with the planning process for a particular task:

Obtain new intelligence reports before planning the sorties for Sector 1.

In terms of execution advice, the user could specify directives such as

Take out targets T1 and T2 on the next raid.

Delay all missions in Schanjok sector by 24 hours.

The first of this pair would lead to the dynamic insertion of appropriate new goals (and actions) at execution time to accommodate the user's request. The second imposes restrictions on the allowed expansion of goals in a given plan at execution time.

10.3 Open-ended Planning

Continuous operation in dynamic environments requires the ability to produce open-ended plans that are tailored to the current information state, but that can grow and evolve in response to events and the acquisition of new information. We use the term *open-ended planning* to describe this capability. Note, however, that it encompasses much more than plan generation. In particular, monitoring is required to test the availability of information and changes in the world. Control is necessary to determine when and how to extend plans in the face of new information. Finally, user involvement is essential to ensure that the open-ended planning process satisfies ever-changing user requirements and preferences.

Section 8 described our work to date on open-ended planning. In particular, we had developed a method for accommodating *abstraction-level planning horizons* that combined identification of when to postpone planning decisions, the establishment of appropriate monitors, and user involvement in the process. Our initial implementation of these ideas was under way, but not yet completed.

In addition to abstract-level planning horizons, we had intended to develop methods for planning for varying temporal horizons. This work would have enabled, for example, generalization from today's rolling three-day model for ATO planning to an approach that enabled a single, integrated plan that extended out in time and depth to limits that made sense given the commander's current knowledge and strategies. Generalized planning horizons of this sort are essential to more flexible planning methods.

10.4 Evaluation

The SIMFLEX agent provides the ability to undertake experiments with CPEF in which different capabilities of the system can be quantitatively evaluated. We had intended to perform experiments in which the ability of different process management and repair strategies could be evaluated for timeliness and effectiveness (in terms of minimizing downstream failures).

10.5 Enriched Domain Models

The ACP KB could be enhanced in several ways both to improve realism and to provide a richer framework for evaluating continuous planning and execution.

Plan Evaluation Criteria During the presentation of our Phase 2 demonstration at IFD 2.0, it was suggested that we identify metrics for plan evaluation that would enable the planning system to generate increasingly 'better' plans through application of a hill-climbing search process. This capability would be valuable to develop, but would require subject matter experts (SMEs) to help articulate appropriate evaluation metrics.

Resource-constrained Planning Detailed resource requirements were intentionally omitted from the ACP KBs used by CPEF. The overall system for which these KBs were originally developed (namely, the ARPI TIE-97-1 system) included a separate resource scheduler (OPIS) that was responsible for both checking resource feasibility during plan generation and performing final resource allocation.

This separation of strategy generation and detailed resource allocation is valuable in that it isolates different but related tasks. However, strong linkage between these two forms of planning is required to ensure that effort is not wasted on detailed strategic plans that are not viable with respect to available resources. To achieve this linkage, we believe that the current ACP KBs should be extended to associate rough estimates of resource requirements with activities. These low-fidelity models would provide a 'sanity check' on the feasibility of plans, without forcing the planning technology to solve the entire resource allocation problem. In addition, an *apportionment* layer could be added that would determine how to allocate resources among high-level goals. The apportionment process would allow generation and comparison of high-level resource allocation strategies, such as keeping large numbers of cruise missiles in reserve.

Phasing In USAF/USN usage, phases are not necessarily strictly sequential. A phase is associated with a high-level goal, like attainment of air superiority or support of a ground war. Phases are loosely sequenced, meaning that phase P1 must start before phase P2, but P1 need not complete fully before P2 begins. Typically, phase sequencing is constrained in part by the requirements of the overall plan (e.g., the need to support a ground or maritime war).

The ACP KBs currently support a notion of phasing that is strictly sequential. Phasing is implemented in a brute-force manner: a complete subplan for phase N is generated before generating phase N+1. This leads to tractability problems for large plans; each phase consists of 10 to 35 planning levels, with plan generation being significantly slower for deeper plans.

A robust and tractable notion of phased plans would lead to more realistic plans. To this end, the strictly sequential nature of phasing in the KB should be relaxed to model actual practice more accurately and to improve tractability.

11 Conclusions

11.1 Summary of Accomplishments

Despite the early termination of the project, much was achieved during its twenty-month lifetime.

The prototype CPEF system that we developed illustrates how AI technologies for reactive control, monitoring, knowledge-based planning, and multiagent organization could be leveraged to produce a continuous planning and execution system for a component of the overall JFACC domain. Central to this system is the use of generic technology for process management, which initiates, controls, and coordinates the activities of the different modules within the system. While this technology is applied within the current version of CPEF to manage plan generation and execution, it applies more broadly to any domain that requires adaptive process management. Extensive documentation for CPEF was developed, including instructions for running the system and several demonstration scenarios.

In developing CPEF, notable technical advancements were made in the areas of models for tracking plan execution, automated extraction of monitors from plans, and generalized models of execution failure and plan repair. Promising work in open-ended planning was begun, but not completed.

Our project contributed much to the program beyond the development of our own technology. We were actively involved with the PlanGen group from the outset and had significant influence on the design of the PlanGen system developed for Phase II. Our project was the only one that integrated with the PlanGen framework, providing both the translation of different courses of action developed within CPEF into the JFACC Plan Server and task-level integration for options generation and plan repairs through the JFACC event channels. Our technology for plan generation and plan repair played a key role within the Phase II demonstration, being used to populate the JFACC Plan Server with multiple courses of action produced through different commander's guidance, and to effect changes to plans in the face of information updates about incorrect assumptions and failed plan execution.

In addition to the collaboration with the PlanGen group, we worked with other Technology Base members within JFACC to formulate the CommonP language. CommonP was developed both to rapidly enable semantically meaningfully interoperability among the TechBase systems, and to influence future iterations of the JFACC Plan Representation to provide better support for automated planning technologies.

11.2 Technology Transfer

Despite the early termination of JFACC, the CPEF system will continue to play a role in DARPA-sponsored research. We are supporting transition of portions or all of CPEF to provide process management technology for several other projects. As mentioned above, the core CPEF technology is general-purpose and domain-independent, which makes it possible to support these transitions with minimal effort.

Dr. Pauline Berry from SRI and Dr. Brian Drabble from the University of Oregon are developing a system called Smart Workflow for ISR Management (SWIM) within the Advanced ISR Management (AIM) program, which will be built on top of CPEF. In the short term, the focus will be on reuse of the execution tracking, monitoring, and simulation capabilities from CPEF. In the longer term, there is a possibility that the plan generation and repair capabilities from CPEF will also be used to enable dynamic synthesis of process management plans.

SRI's main effort within the Small Unit Operations (SUO) program is developing monitoring and execution capabilities based on CPEF modules. A smaller SUO effort at SRI, led by Dr. David Wilkins, has proposed to use CPEF for the development and continuous management of plans to support ongoing coordination of small units.

11.3 Problems and Obstacles

Our efforts within the program were hampered by several factors, which limited the technical accomplishments that we were able to achieve. We briefly discuss several of these issues, in the hope that they may be of use in the formulation of the follow-on JFACC program.

Misperception: Process Management *vs.* Plan Generation The combination of the ability of our system to produce plans and the need for plans to be generated within the program resulted in a tendency to categorize our project as a 'planning effort'. While automated plan generation is one component of CPEF, it was not the intended focus of the project. Rather, we sought to develop the process management capabilities necessary to support the type of continuous operations inherent to the JFACC domain.

This misperception was exacerbated by the fact that Phase II of the program focused on activities related to creation of a single plan rather than continuous plan development. Furthermore, it neglected operating dynamics and execution issues almost entirely. In particular, runtime adaptations were added to the Phase II demonstration as an afterthought only weeks prior to IFD 2.0. No simulation environment was provided, nor were appropriate alternatives (such as scripted event generation) defined to enable demonstration of the ability of systems to adapt their behavior continuously in response to unforeseen events or partial execution results.

Another reason for the low visibility of our process management capabilities was the nature of the plans that we were directed to produce for the demonstration, which focused on refining air objectives to the level of an ATO. The significance of these plans is sufficient that it was necessary to refine objectives completely down to this level. As such, we were unable to demonstrate how CPEF's process management technology could be used to dynamically refine activities in real time. This kind of dynamic update would be more appropriate in updating (for example) a logistics support plan for moving materials into theater to support an ATO.

CPEF *vis a vis* JFACC System Integration CPEF was a problematic fit with the system integration plan envisioned for the program. As mentioned in Section 1, CPEF provides a microcosm of the types of process management capabilities required for JFACC, being limited to the management of continuous planning and execution for air objectives. As such, CPEF provided a vertical slice through the set of services that were to be provided by Workflow, PlanGen, and several specialized component technologies. Thus, while CPEF did not provide breadth of coverage for the JFACC domain, it showed how technologies can be used to manage 'living plans' within highly dynamic worlds.

Demonstrations and Integration *vs.* Technology Development Almost from the outset of the program, we were redirected from our initial project plan to focus on building a baseline demonstration system for Phase II. We played a key role in the Phase II demonstration system, serving as the sole source of plans deposited into the JFACC Plan Server. Furthermore, we completed an initial task-level integration of CPEF into the PlanGen framework that supported requests to generate multiple courses of action and to repair plans in the face of unexpected events or violated assumptions.

While we expected to contribute to program meetings and provide technology for use in program demonstrations, the percentage of our time devoted to these efforts was significantly higher than anticipated. As a relatively small Technology Base project, fulfilling these responsibilities (with the high levels of travel and documentation involved) consumed a significant part of our resources through the Phase II demonstration. The result was a lack of available time for developing our proposed technological advances.

11.4 Closing Remarks

The JFACC program sought to develop a technological framework to support the creation, adaptation, and use of *living plans* that would grow and evolve over time in response to updated knowledge of the operating environment, tasking changes, and unanticipated events in the world. Our project focused on several core issues that are central to providing such management for living plans. We developed innovative capabilities in the areas of execution tracking, plan adaptation, monitoring, generalized failure models, and process management for continuous planning. In addition, we built the prototype CPEF system which showed how these capabilities could be used to create and execute sophisticated air campaign plans in highly dynamic operating environments. We believe that these achievements constitute significant progress toward the objective of truly continuous planning and execution for complex and unpredictable domains.

12 Bibliography

References

[1] J. F. Allen. Recognizing intentions from natural language utterances. In *Computational Models of Discourse*, pages 107–166. MIT Press, Cambridge, MA, 1983.

[2] J. L. Ambite and C. A. Knoblock. Planning by rewriting: Efficiently generating high-quality plans. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.

[3] K. Erol, J. Hendler, and D. S. Nau. Semantics for hierarchical task-network planning. Technical Report CS-TR-3239, Computer Science Department, University of Maryland, 1994.

[4] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, and J. McGuire. Specification of the KQML Agent-Communication Language. Technical Report EIT T R92-04, Enterprise Integration Technologies, Palo Alto, CA, 1992.

[5] T. Garvey and K. Myers. The intelligent information manager. Final Report SRI Project 8005, Artificial Intelligence Center, SRI International, Menlo Park, CA, 1993.

[6] M. P. Georgeff and F. F. Ingrand. Research on procedural reasoning systems. Final Report Phase 1, Artificial Intelligence Center, SRI International, Menlo Park, CA, October 1988.

[7] M. P. Georgeff and F. F. Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989.

[8] F. F. Ingrand, J. Goldberg, and J. D. Lee. SRI/GRUMMAN Crew Members' Associate Program: Development of an authority manager. Final Report SRI Project 7025, Artificial Intelligence Center, SRI International, Menlo Park, CA, March 1989.

[9] S. Kambhampati and J. Hendler. A validation-structure-based theory of plan modification and reuse. *Artificial Intelligence*, 55(2):192–258, 1992.

[10] T. J. Lee. The air campaign planning knowledge base. Technical report, Advanced Automation Technology Center, Menlo Park, CA, 1998.

[11] D. L. Martin, A. J. Cheyer, and D. B. Moran. The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13:91–128, 1999.

[12] D. B. Moran, A. J. Cheyer, L. E. Julia, D. L. Martin, and S. Park. Multimodal user interfaces in the Open Agent Architecture. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces (IUI97)*, Orlando, FL, 6-9 January 1997.

[13] K. L. Myers. *The ACT Editor User's Guide.* Artificial Intelligence Center, SRI International, Menlo Park, CA, 1993.

[14] K. L. Myers. *User's Guide for the Procedural Reasoning System.* Artificial Intelligence Center, SRI International, Menlo Park, CA, 1993.

[15] K. L. Myers. Advisable planning systems. In A. Tate, editor, *Advanced Planning Technology.* AAAI Press, Menlo Park, CA, 1996.

[16] K. L. Myers. Strategic advice for hierarchical planners. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96).* Morgan Kaufmann Publishers, 1996.

[17] K. L. Myers. Abductive completion of plan sketches. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence.* AAAI Press, 1997.

[18] K. L. Myers. Towards a framework for continuous planning and execution. In *Proceedings of the AAAI 1998 Fall Symposium on Distributed Continual Planning*, Menlo Park, CA, 1998. AAAI Press.

[19] B. Nebel and J. Koehler. Plan reuse versus plan generation: A theoretical and empirical analysis. *Artificial Intelligence*, 76:427–454, 1995.

[20] L. Pryor and G. Collins. Planning for contingencies: A decision-based approach. *Journal of Artificial Intelligence Research*, 4:287–339, 1996.

[21] D. E. Wilkins. Recovering from execution errors in SIPE. *Computational Intelligence*, 1(1):33–45, 1985.

[22] D. E. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan Kaufmann, 1988.

[23] D. E. Wilkins. Can AI planners solve practical problems? *Computational Intelligence*, 6(4):232–246, 1990.

[24] D. E. Wilkins and R. V. Desimone. Applying an AI planner to military operations planning. In M. Fox and M. Zweben, editors, *Intelligent Scheduling.* Morgan Kaufmann, 1994.

[25] D. E. Wilkins and K. L. Myers. A common knowledge representation for plan generation and reactive execution. *Journal of Logic and Computation*, 5(6):731–761, December 1995.

[26] D. E. Wilkins and K. L. Myers. A multiagent planning architecture. In *Proceedings of the Fourth International Conference on AI Planning Systems*, 1998.

A CPEF Component Technologies

A.1 Procedural Reasoning System (PRS)

PRS is a framework for constructing knowledge-based reactive controllers that can operate as embedded systems within highly dynamic environments [7]. These reactive controllers can perform actions to accomplish explicitly assigned tasks, while providing real-time response to unexpected events that result from factors that lie beyond the system's control. PRS technologies have proven useful in developing several demanding applications that required integration of reactive and goal-oriented behavior, including real-time tracking [5], a monitoring and control system for the Reaction Control System of the NASA Space Shuttle [6], and a control system for naval battle management aboard a Grumman E-2C [8].

Individual instantiations of PRS are referred to as PRS *application agents.* A PRS application agent consists of a database containing current beliefs or facts about the world, a set of current goals, a set of predefined *procedures* (encoded in the Act representation language [25]) describing how sequences of actions and tests may be performed to achieve certain goals or to react to particular situations, and *intentions* that keep track of the current procedures being executed by the agent. An interpreter manipulates these components, by selecting appropriate procedures for execution based on the system's beliefs and goals, then creating the corresponding intentions, and finally executing them.

A PRS agent interacts with its environment through its database (which acquires new beliefs in response to changes in the environment) and through the actions that it performs as it carries out its intentions. While the system is running, it constantly monitors incoming information and goals. The predefined procedures are activated in response to the adoption of a new goal or to some change in the world. This combination of goal- and data-driven activity yields a flexible, adaptive execution framework. In particular, any intention can be interrupted and reconsidered in the light of new information about the world. The monitoring method used guarantees that any new fact or goal is noticed in a bounded time, thus providing rapid response to new events.[7]

Multiple PRS agents can be active simultaneously. Each agent has its own local goals, intentions and database, and runs asynchronously in the overall framework. A message-passing facility enables communication among agents to support parallel, distributed problem-solving.

PRS procedures support a broad range of action types, ranging from testing conditions to achieving goals and waiting for events. In terms of control, the procedures provide conditional branching, iteration, recursion, and parallelism. Procedures are not limited to describing activities in the external world, but can also manipulate the internal beliefs, goals, and intentions of PRS. Such *metalevel procedures* can encode actions that influence the operation of the system itself, such as methods for choosing among multiple applicable procedures, modifying

[7]The bound on the cycle time is not absolute, but rather depends on the time required to execute the basic actions in a given domain.

intentions, or computing the amount of reasoning that can be undertaken given the real-time constraints of the problem domain.

PRS is an excellent foundational technology for process management and control within CPEF: it is reactive, integrates goal-driven and event-driven activities uniformly, and has proven effective in numerous applications. The ability to define multiple PRS agents supports the simultaneous use of multiple instantiations of the abstract agent model, thus enabling cooperative problem-solving by distributed agents.

A.2 SIPE-2

SIPE-2 is a state-of-the-art *hierarchical task network* (HTN) planning system that supports partial-order planning at multiple levels of abstraction. 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 assigned goals. Given an arbitrary initial situation, the system either automatically or under interactive control combines operators to generate plans to achieve the prescribed goals.

SIPE-2 has the ability to generate action sequences that include parallel actions, conditional actions, and resource assignments. It provides powerful temporal reasoning capabilities, supporting the use of any of the 13 Allen temporal relations [1] or qualitative constraints between the endpoints of any pair of actions in an operator. SIPE-2 also supports conservative replanning: minimal modifications to plans that enable them to be adapted when the world in which they are being executed changes in unexpected ways. SIPE-2 provides a powerful graphical user interface to aid in generating plans, viewing complex plans, and following and controlling the planning process. In contrast to most AI planning research, heuristic adequacy (efficiency) has been one of the primary goals in the design of SIPE-2, yielding a system that is capable of solving large, real-world problems.

The SIPE-2 technology is generic and domain-independent, and 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, containing oil spills, travel planning, construction tasks, producing products from raw materials under production and resource constraints, and joint military operations planning [23, 24]. Of greatest relevance to JFACC is the use of SIPE-2 as the core planning technology for ARPI's fourth Integrated Feasibility Demonstration (IFD-4), where it was used to generate high-level strategies for Air Campaigns.

A.3 Advisable Planner

With most AI planning systems, user control over the planning process is limited to the specification of the top-level goals that the system is to solve. Certain more advanced planners support additional user control by enabling both the selection of operators and instances for variables at planning time, and the ordering of the goal expansion process. These interactions are too fine-grained for most users, however, who want to be involved with the planning process at a higher, more strategic level.

The *Advisable Planner* enables users to direct the planning process in a more powerful and general manner, through the provision of high-level guidance that impacts the nature of the solutions generated. The Advisable Planner supports two main forms of user guidance: *strategic advice* and *plan sketches*. Strategic advice constitutes recommendations on how tasks are to be accomplished, in terms of both approaches and entities to be used. Plan sketches provide a second form of advice through the specification of individual tasks that are to be included in the overall plan.

To illustrate the value of advice, consider a travel-planning domain. A typical planner might allow a user to sketch a high-level outline of a trip, specifying information such as which locations to visit at what times and for what overall cost. The planner would then fill in the appropriate details. Most individuals, however, want to influence their itineraries to a much greater extent, specifying details such as the modes of transportation for various legs, individual carriers to use, accommodation requirements, and restrictions on costs for various aspects of the trip. Advice provides the means by which users can generate customized plans in this way.

The Advisable Planner consists of a module for advice processing and enforcement that is layered on top of the SIPE-2 planning system. A detailed discussion of the Advisable Planner concept can be found in [15]. Additional technical background on advice and advice enforcement can be found in [16, 17].

A.4 Multiagent Planning Architecture (MPA)

The Multiagent Planning Architecture (MPA) [26] is an agent-based framework for integrating diverse technologies into a distributed system capable of solving complex planning problems. MPA was designed explicitly to address planning problems that cannot be solved by individual systems, but rather require the coordinated efforts of a diverse set of technologies and human experts. Agents within MPA share well-defined, uniform interface specifications, making it possible to explore a broad range of cooperative problem-solving strategies. MPA agents can be sophisticated problem-solving systems in their own right, and may span a range of programming languages. MPA facilitates rapid incorporation of new tools and capabilities through a combination of its open design and wrapper technologies that enable easy encapsulation of legacy software systems as agents.

MPA is organized around the concept of a *planning cell*, which consists of a collection of agents committed to a particular planning process. A planning cell contains two distinguished agents — the *planning-cell manager* and the *plan server*. The *planning-cell manager* composes a planning cell from a pool of available agents and distributes the planning task among the selected agents. The *plan server* is the central repository for plans and plan-related information during a planning session. It accepts incoming information from planning agents

(PAs), performs necessary processing, stores relevant information, and makes this information accessible to any PA through queries. Within MPA, notions of *baselevel planning cells* and *metalevel planning cells* have been defined, where the baselevel cells provide sequential solution generation and the metalevel cells employ baselevel cells to support parallel generation of qualitatively different solutions. The use of metalevel cells thus enables rapid exploration of the solution space for a planning problem.

The MPA architecture rests upon a significant amount of infrastructure. One component is a *shared plan representation* that can be understood by all planning agents. MPA employs the Act formalism for this purpose [25, 13]. Additional components include a communication substrate to support asynchronous interagent message passing across networks, and tools to facilitate the construction of agents and planning cells. MPA provides both a message format and a message-handling protocol to support the sharing of knowledge among agents involved in cooperative plan generation.

MPA is distinguished from other agent architectures (such as the Open Agent Architecture [11, 12]) in its emphasis on application to large-scale planning problems. The framework includes agents designed specifically to handle plans and planning-related activities. In addition, interagent communication protocols are specialized for the exchange of planning information and tasks. Another distinguishing feature of MPA is the emphasis on facilitating the integration of agents that are themselves sophisticated problem-solving modules. Most agent architectures develop specialized agents that are suited for operation within that specific architecture rather than incorporating legacy systems.

The MPA framework has been used to develop several large-scale problem-solving systems for the domain of Air Campaign Planning (ACP). One such application integrated a set of technologies that spanned plan generation, scheduling, temporal reasoning, simulation, and visualization. These technologies cooperated in the development and evaluation of a complex plan containing more than 4000 nodes. This integration validated the utility of MPA for combining sophisticated stand-alone systems into a powerful, integrated problem-solving framework.

B Preliminary Demonstrations

B.1 Baseline Demonstration (December, 1997)

In early December (as part of IFD 1.7), SRI delivered and demonstrated its initial CPEF prototype. The delivered system provided a baseline framework illustrating how SRI's planning reactive control technologies can support the continuous development, monitoring and adaptation of Air Campaign plans. In particular, the system demonstrates the following technical capabilities, within the context of force application for air superiority objectives:

- Fully automated and interactive plan generation
- User-guided plan development through the provision of advice
- Situation monitoring for both user-generated and automatically generated monitors
- Execution tracking
- Automated adaptation of plans in response to updated situation information
- Automated adaptation of plans in response to unexpected problems during plan execution

Given the short time in which this system was developed, the technical capabilities were somewhat shallow and brittle. Nevertheless, this initial demonstration system successfully illustrated the value that AI planning and reactive control technologies can contribute to the automation of key JFACC capabilities.

This initial demonstration used the Granola scenario developed for the ARPI TIE-97-1 system [10]. Extensive documentation for the demonstration (including directions for loading and running CPEF) is available on the web at the URL

```
http://www.ai.sri.com/~cpef/jfacc/acp-demo-dec97.html
```

B.2 Phase 2 Demonstration System (June, 1998)

During IFD 2.0, SRI delivered version 1.6 of the CPEF system for use within the JFACC Phase 2 demonstration. The delivered CPEF system met all software compliance requirements, documentation requirements, and delivery standards imposed on Technology Base projects by the TIED (Technology, Integration, Evaluation, Demonstration) group.

B.2.1 Overview

CPEF's role within the overall Phase 2 demonstration system was to

- perform rapid generation of alternative courses of action based on specifications of different sets of user guidance
- identify and monitor for situational changes whose violation should trigger plan repair
- undertake plan repair in response to detection of monitored conditions

For this demonstration, CPEF's operational scope was limited to the planning of air superiority objectives down to the ATO level, and then managing repairs and adaptations of those plans (both prior to and during execution of the plan) in response to results from plan execution and unexpected changes in the world.

This version of CPEF was integrated into the overall JFACC system through two means. First, JFACC *event channels* were used to provide communication links between CPEF and the PlanGen framework. The integration was limited to (a) servicing requests for the generation of options, and (b) repairing plans in the face of situation updates. Second, SRI developed a translator for mapping CPEF-generated plans (stored in SRI's Act representation language) into the C2 Schema. Through a combination of the translator and the event channel connection, CPEF-generated plans and plan repairs could be automatically installed into the JFACC Plan Server, and displayed using the JFACC Visualization services.

Complete documentation for the delivered CPEF system is available from the CPEF web site, including operating instructions, detailed descriptions of the system's capabilities, and sample plans (see the URL `http://www.ai.sri.com/~cpef/jfacc/ifd2.html`). Here, we provide a brief overview of key concepts and contributions.

We note that this demonstration did not make direct use of the simulation capabilities within SIMFLEX, primarily due to time limitations that had been imposed.

B.2.2 Operational Summary

The plans in this demonstration are derived through strategy-to-task refinement of objectives for defensive and offensive air superiority. Refinement of objectives proceeds through 17 levels, terminating at the level of the ATO. The final plans contain several thousand nodes and require less than a minute to produce.

The knowledge bases within CPEF support several high-level strategies for achieving air superiority objectives. The *advice* capabilities within CPEF enable users to instruct the system on the kinds of strategies to be employed when exploring different options. The main plan used within the demonstration employs a 'rollback and defend' strategy for attaining defensive air superiority. It involves initiation of preemptive attacks against all threats that are "close" to Blue COGs, with simultaneous establishment of CAPs to defend those COGs. Air operations at threatening Red airbases are disrupted by attacking their constituent runways. The strategy for offensive air superiority involves establishing a breach of Red IADS in the northeast corner of West Cyberland (referred to as the *Sentani* sector in the CPEF representation of the Cyberland geography). This breach will both support strikes against front-line and follow-on invading forces, and enable subsequent attacks against lines of communication (LOCs) supporting the invasion. The offensive component of the plan aims to deny the enemy its air picture by inflicting 'blindness' in the breach sector, through attacks on the EW/GCI radars for the sector. The plan also includes objectives for extending air dominance through the initial breach sector. However, the demonstration plans omit detailed expansion of these objectives (to reduce the size and complexity of the plans).

During plan execution, an intelligence report is received indicating the possible presence of SA-11 SAMs in the Sentani sector. The JFACC decides to alter the current plan to attack command, control, and communication (C3) facilities that would support the SAMs to further degrade their effectiveness. (These attacks can be conducted almost immediately, without determining the location of the SA-11s themselves.) CPEF performs dynamic repair of the plan currently under execution to incorporate missions to satisfy the decision made by the JFACC.

C Final Demonstration: Overview and Operating Instructions

C.1 Introduction

CPEF (the Continuous Planning and Execution Framework) is a multiagent system that tightly integrates generative planning and reactive plan execution capabilities. This document provides detailed instructions for running a demonstration of CPEF that was developed for the Phase II demonstration of the JFACC program. The plans in this demonstration focus on attaining offensive and defensive air superiority within the JFACC Cyberland scenario.

The demonstration illustrates how CPEF can be used to generate multiple courses of action under user guidance, track execution of selected plans, and adapt plans as necessary in response to changes in the situation or unexpected execution results. Key technical capabilities illustrated by this CPEF demonstration include

- User-guided automated refinement of objectives (via *advice*)
- Automated extraction of monitors from generated plans
- Managed tracking of plan execution
- Adaptation of plans in response to situation changes and progress through plan execution

Section C.2 summarizes the overall demonstration, from both a technical and an operational perspective. Section C.3 presents a system-level view of CPEF, while Section C.4 describes how to load CPEF. Section C.5 provides detailed instructions for running the demonstration. Section C.6 contains a list of helpful hints for demonstrators, including tips for individuals who are not familiar with LISP environments.

C.2 Demonstration Overview

C.2.1 Technical Summary

Figure 5 provides a high-level summary of the demonstration. The timeline along the bottom of the diagram shows exogenous events that are outside of the system's control. Above that, activities are organized along three functional roles: the User, the Planner, and the Executor. The system is designed so that the User plays an active role in the plan development and execution processes. The term Planner is used generically to refer to planning-related activities: plan generation, plan analysis, plan repair. The Executor provides three main threads of activity in parallel: *situation monitoring*, *execution tracking*, and *plan management*. Situation monitoring refers to the process of checking for key changes in the world state that

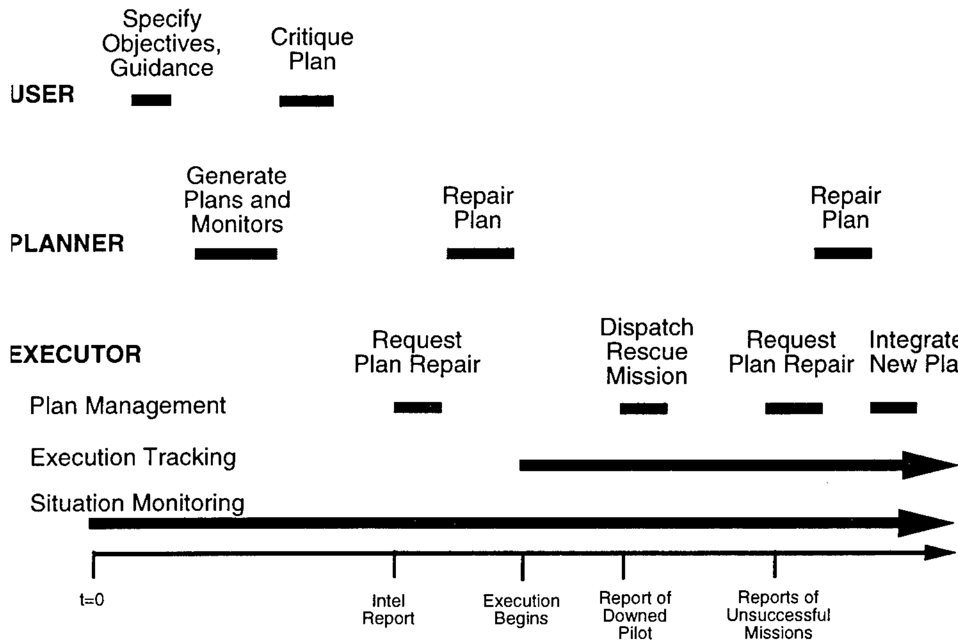

Figure 5: Demonstration Timeline

could impact plans or ongoing planning processes. Execution tracking involves monitoring the progress through execution of chosen plans, to ensure that the process is proceeding in an acceptable manner. Plan management refers to the overseeing of the modification of plans in response to changing situation information and incoming results from execution results.

The demonstration begins with the User specifying air objectives for a given campaign along with advice or guidance as to the kind of plan(s) that the User would like produced. The Planner, in accord with stated guidance, generates a plan to satisfy those objectives relative to its current knowledge of the operating environment. The completed plan is reviewed by the User who can recommend changes to satisfy any outstanding concerns. The Planner then produces an updated plan that incorporates the User's feedback. As planning proceeds, the Executor monitors the environment for events that are relevant to the developing plan. Receipt of an intelligence update that invalidates parts of the plan prompts a request to replan the affected portions of the plan. Monitoring of the environment continues as execution of the plan commences. In addition, the Executor tracks progress through the execution of the plan to determine whether modifications are needed in response to the status of the execution, and undertakes such modifications as required.

C.2.2 Operational Overview

The plans in this demonstration provide strategy-to-task refinement of defensive and offensive air superiority objectives, terminating at the level of the Air Tasking Order (ATO).

The knowledge bases within the system support a number of high-level strategies for achieving air objectives. The user can designate the use of different strategies through provision of appropriate pieces of advice to the system.

This initial air superiority plan generated within the demonstration calls for the weight of effort to focus on the Sentani sector in northeast West Cyberland. Within the JFACC scenario, it is this sector from which the ground attack toward the oil fields was launched. Sentani sector contains lines of communications and follow-on forces supporting the attack. Since the ground war will call for many sorties to be flown into this sector (perhaps urgently to support the retreating Blue forces), the JFACC decides to breach the IADS there, even though the defenses are relatively strong. In addition to supporting the ground war, choosing Sentani as the primary point of effort supports the naval objective of controlling the sea lines of communication on the north side of the island.

To conserve striking power for the Sentani sector, a nonpreemptive defense scheme is chosen to protect Blue centers of gravity from Red strike threats. The particular tactic selected is to put up barrier CAPs along the entire border. This has the advantage of defending against relatively many threats in relatively few places (rather than, say, flying CAPs over all friendly COGs). Also to conserve striking power, SOF and jamming assets are used to deny the enemy its air picture over Sentani. Two early-warning radar sites that cover Sentani are deemed to be vulnerable to SOF attacks, as they are isolated and relatively unprotected; they are attacked in this manner.

A second plan is created that is more offensive in nature. It calls for strikes on runways to disrupt the operational tempo of the enemy's offensive air war. Though it soaks up sorties that could be used to support the ground war directly, it is felt to be lower risk as it considerably reduces enemy flexibility. For this reason, the user elects to go with this plan.

Prior to the commencement of plan execution, national assets have tracked the delivery of a small shipment of Exocet missiles to West Cyberland. Early intelligence reports predicted that the missiles were headed south, to the Nojimsan airbase. However, late-breaking intelligence has definitively located the missiles at Frans Kaisiepo airbase in the north, within striking range of the Kennedy carrier. It is felt that an attack on the Kennedy is imminent, to relieve the pressure exerted on the invaders by carrier-based aircraft, and to possibly cause a political backlash in the United States. This threat must be incorporated into the existing air superiority plan, so a plan repair is initiated.

Plan execution then commences. At some point during the execution, notification is received that a pilot has been downed. The system responds by instigating an appropriate activity (such as dispatching a Search and Rescue mission).

Due to poor weather over Sentani during the early phase of the Air Campaign, many sorties are unable to fly against targets in that sector. To better utilize available assets, the initial attacks are extended westward to the Rockatoon sector, where weather is somewhat better. This has the advantage of opening the Red capital to (nonstealth) attacks earlier, disrupting national C2 and logistics centers, and in general forcing Red to defend in more places.

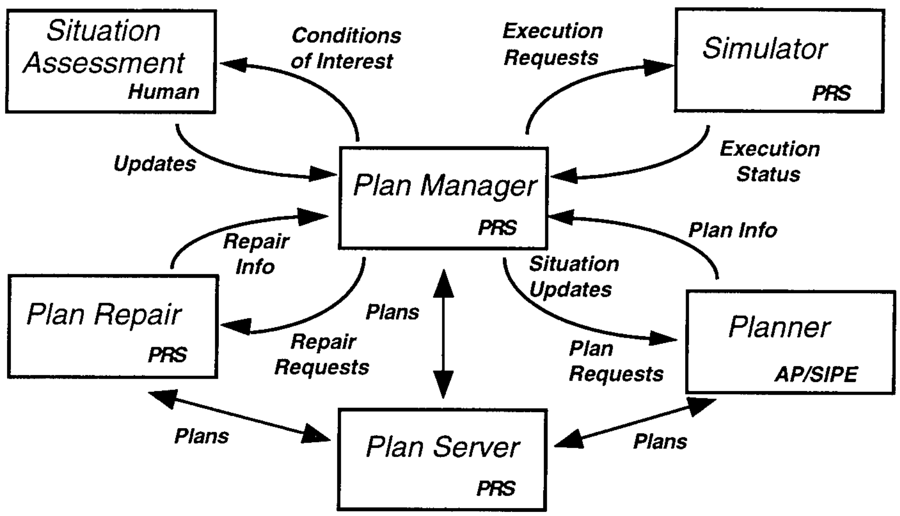

Figure 6: Architectural Overview of CPEF

C.3 System Organization

CPEF is an integrated planning and execution system composed of a number of AI technologies (see Figure 6). Most relevant are the SIPE-2 generative planner, the Advisable Planner, which is a module layered on top of SIPE-2, that enables user advisability of the automated planning process, the Procedural Reasoning System (PRS) for multiagent reactive control, and the Act Editor for editing and viewing declarative representations of planning knowledge (called *Acts*).

Within this demonstration, SIPE-2 and the Advisable Planner provide the advice specification, core planning, and plan repair capabilities. PRS plays a number of different roles. First, it invokes and tracks execution of plans and standard operating procedures. Second, it performs all situation monitoring. Third, it manages the planning process, determining when to generate plans or to instigate plan repairs. Finally, PRS is also used to provide simulation capabilities within CPEF.

To manage these various roles, three different PRS agents have been created within CPEF, named PlanMan, Plan Repair, and SIMFLEX. The PlanMan agent provides tracking of plans and situation monitoring. The Plan Repair agent mediates interactions with SIPE-2. The SIMFLEX agent provides simulation of plan execution (SIMFLEX is derived from *Sim*ulated *Fl*exible *Ex*ecution). A fourth agent, the PlanServer, provides a repository for plans and options that have been created. These agents communicate with each other through the asynchronous exchange of messages using the communication protocols of the Multiagent Planning Architecture (MPA).

C.4 Loading CPEF

The first step to loading CPEF is to load MPA. In the directory `~cpef/released/scripts/`, invoke the following command from the UNIX shell.

```
init-jfacc-agents
```

This command will launch an initialization program called StartIt, from which you can launch the CPEF agents. The StartIt window contains a control panel for each of the CPEF agents, along with two MPA agents for providing *agent namer services*, namely ANS Facilitator and ANS Monitor. These latter two agents manage communication within the system and are not used directly within CPEF.

First, start the ANS Facilitator by clicking on its control panel and selecting the Start option.

Next, start the ANS Monitor by clicking on its control panel and selecting the Start option.

Then, start each of the three CPEF agents in the same manner (Executor, Planner, PlanServer). When launching CPEF agents, an Emacs window will appear followed by the graphical user interface for that component. When the interface appears, invoke the following command in the LISP Listener window for each agent:

```
(init-jfacc-demo)
```

An MPA Trace window for each CPEF agent will appear that displays message traffic between agents. These windows can be iconified, since they are not important for the demonstration.

For each of the PRS agents (PlanMan, PlanRepair, SIMFLEX) running within the Executor, a separate PRS trace window is created. These windows display information about the activities in which the agents are engaged. If you wish to position these ahead of time, go to the `Application` menu for the interface to the Executor and select `PRS`. The trace windows will appear and should be positioned appropriately.

C.5 Running the Demonstration

The demonstration involves interacting with several of the subsystems within CPEF.

C.5.1 Overview of Planner and Planning Knowledge

From the Application menu, select SIPE-2. The Planner in this demonstration is a combination of SIPE-2 with the Advisable Planner module. SIPE-2 provides a range of tools for displaying its constituent knowledge bases: the domain model, operator base, advice, and currently defined problems. These tools are accessible by selecting the `DRAWINGS` menu.

For demonstrations to audiences who are unfamiliar with automated planning technology, it is recommended that a simple overview of the various knowledge bases be provided. Select the `Problem` option on the `DRAWINGS` menu. A list of the available problems to be solved will appear. This demonstration employs `AIR-SUPERIORITY-A`.

The green hexagonal node in the middle of the displayed problem constitutes the top-level task to be solved for this problem. (Information on interpreting the graphical representations of plans within SIPE-2 can be found at the end of this section.) Clicking on that node will pop up more detailed information. Of the displayed slots, the most relevant is

```
Goals:   (AS PHASE-I D+0)
```

which indicates the goal to be achieved within this problem. In particular, the goal is to generate a plan to achieve air superiority during Day 0 of Phase I.

Next, select the `Operator` menu item, then `Menu of Operators`. A list of the operators defined for the domain will be presented. Feel free to show a range of operators. Specific recommendations are

```
AIR-SUPERIORITY-FOR-PHASE
ACHIEVE-OFFENSIVE-AIR-SUPERIORITY
ACHIEVE-DEFENSIVE-AIR-SUPERIORITY
```

These operators are used at the outset of the plan generation process.

Select the `World` option on the `DRAWINGS` menu to display the world model that has been loaded into the planner. In the resulting pop-up menu, select `Display all Predicates` to print a full summary of all predicates in the domain. The domain includes predicates that model a range of concepts from basic background information (borders, adjacency information) to intelligence information (threat axes, threat levels, COGs). Information about capabilities of various assets is stored in a separate class hierarchy structure.

Advice is used within the system as a way of declaring preferences for the kinds of plans that a user wants to generate. Several predefined sets of advice have been loaded with the domain, although advice can also be specified interactively. By employing different sets of advice, the user can guide the planning system to produce qualitatively different plans.

Click on the `Advice` menu and select `Active Advice` to see a listing of the currently defined advice within the system. With the default settings, a textual summary of the active advice will be presented. Interesting examples to highlight include

Use F-15Cs to man all point-defense CAPs over airfields
Disable SAMs by attacking launchers directly
Neutralize enemy intercept capability by denying them their air picture

Notes on SIPE-2 displays Below is information for interpreting the graphical representations of SIPE-2 plans:

- Circular nodes designate either the start or end of a plan.
- Hexagonal nodes with green boundaries correspond to goals to be planned.
- Rectangular nodes with blue borders correspond to primitive actions.
- Square nodes with black borders correspond to preconditions that must be satisfied.
- Internode arcs represent temporal sequencing constraints (nodes may include additional temporal constraints).
- Nodes with multiple outgoing arcs denote splits (i.e., AND parallelism).
- Nodes with multiple incoming arcs denote joins.
- Clicking on a plan node will pop up a window that provides detailed information for the node.

C.5.2 Plan Generation

SIPE-2 supports both fully automated and interactive advisable planning. For the demonstration, we recommend beginning with interactive planning to facilitate explanation of the planning process. After planning a few levels, automated planning should be invoked to complete the planning process. Throughout interactive planning, the system will highlight individual nodes with red as they are refined, flashing once for each operator considered for the node.

To begin planning, the user specifies advice to guide the planning process. Select the `ADVICE` menu. Choose `Select Scenario` and select the advice scenario `JFACC-PLAN4` to load a predefined set of advice into the system. Additional advice can can be created through the use of the MasterMind tool from the University of Southern California/Information Sciences Institute (USC/ISI).[8] To use MasterMind, select `Create Advice` and choose `MasterMind`. Create the following two pieces of advice.

Breach IADS at SENTANI-SECTOR.
When defending against any STRIKE-THREAT use BARCAP.

[8] Shorter versions of the demo can be given but require using different advice scenarios. To skip the creation of advice using MasterMind, select the advice scenario `JFACC-PLAN5`. To skip both the use of MasterMind and the changing of advice, select advice scenario `JFACC-PLAN6`.

The long window on the bottom of the MasterMind interface contains alternatives that can be chosen to create advice in accord with domain-specific grammars that have been loaded into the system. With each choice that is made, the window will be refreshed to display subsequent choices allowed by the grammars. For example, for the first piece of advice listed above, select `Breach IADS at`; in the ensuing window, select `SENTANI-SECTOR`. Once the advice is completely specified, you can optionally specify a symbol as a label for the advice, then click on `Apply`. Doing so will create the corresponding advice structure and activate it for use. To create additional advice, click on `Clear` to return to the set of initial choices for creating advice. When done specifying advice, click on `Quit` to return to the planner interface. Now select `Active Advice` and note that the newly created advice has been activated.

To begin planning, select the `PLAN` menu. Next, select the menu item `interactive-1`. When prompted for the problem to solve, use the presented value `Air-Superiority-A`. This command causes SIPE-2 to generate a more refined version of the plan by applying an operator to each goal within the current plan. The next plan produced consists of two goals: `(GAIN-DAS PHASE-I D+0)` (i.e., gain defensive air superiority) followed by `(GAIN-OAS PHASE-I D+0)` (i.e., gain offensive air superiority). Again, click on the individual nodes to display additional information.

Click on `continue-1` to continue planning. The next level produces a plan with three nodes:

```
(DEFEND-AGAINST-ALL-THREATS D+O LOW)
(BREACH-IADS D+0 LOW)
(EXTEND-AIR-SUPERIORITY PHASE-I D+0 MED)
```

Click on `continue-1` to generate the next level. At this stage, certain nodes will have pink text, indicating that decisions related to these nodes have been impacted by the advice in place for the demonstration. Various commands on the `Advice` menu enable further information related to the effects of advice:

- Click on `Node Impact (T)`, and then select a plan node. The system will print a textual summary of advice that has influenced decisions for this node (i.e., (T) stands for Textual).

- Click on `Advice Impact (G)`, and then select a piece of advice. The system will highlight nodes that have been impacted by that advice, at any level in the plan development process ((G) stands for Global).

Continue plan development by selecting `continue-1` from the `PLAN` menu. At the fourth level down, the overall structure of the plan will become apparent (see Figure 7). At this point, the remainder of the plan should be completed automatically by selecting the `continue` command. From the list of options, select `Finish Planning Automatically` and click `Do it` on the next two pop-up windows.

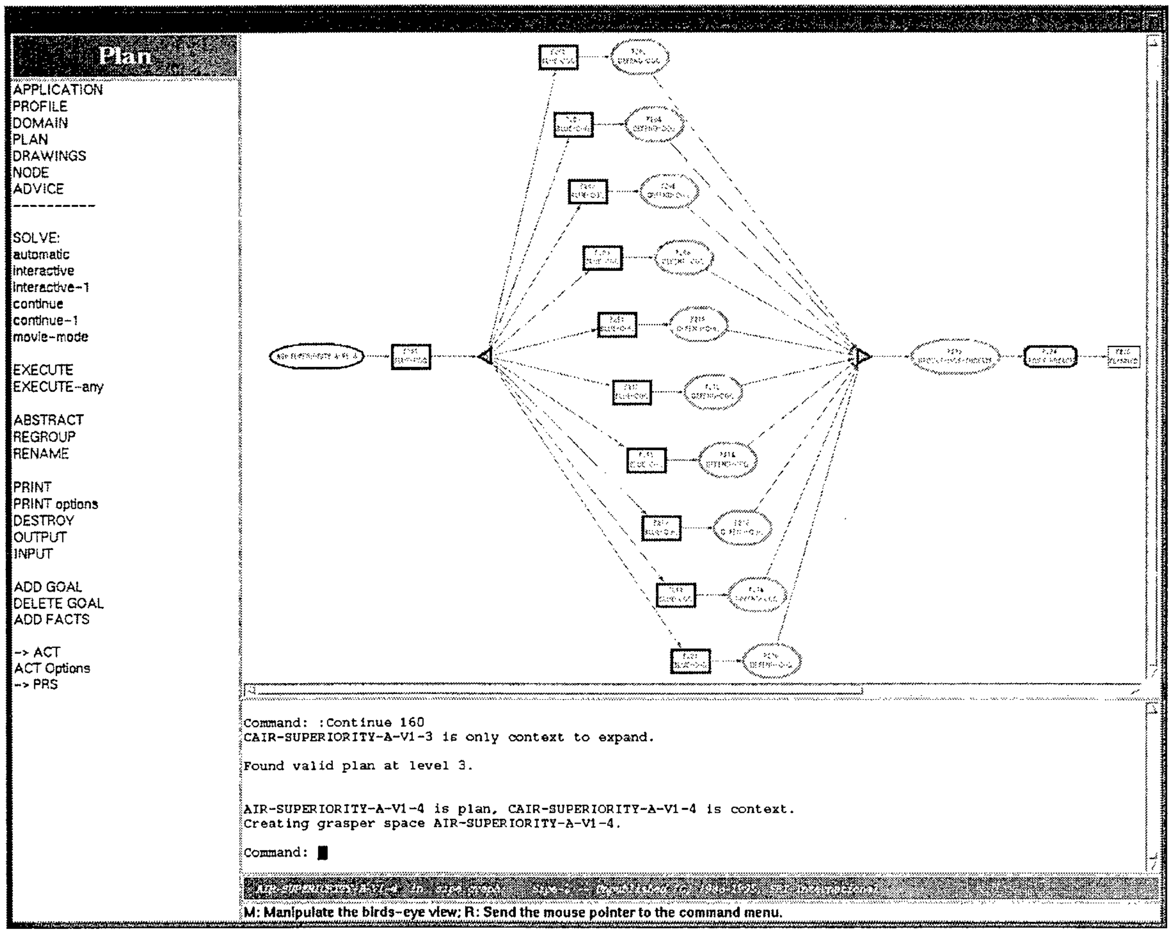

Figure 7: Initial Plan at the Fourth Level of Refinement

The planner will take approximately 30 seconds to complete, yielding a plan at level 17. Figure 8 provides a ‘birdseye view’ of the initial plan at low resolution (the birdseye view is displayed automatically by the system).

Now, demonstrate the ability to incrementally modify a plan by changing the defined advice. In the `ADVICE` menu, select `Change Advice`. Unhighlight `:NON-PREEMPTIVE-DEFENSE` and highlight `:PREEMPT-RUNWAYS` and `:MASS-VS-AIRBASES`. SIPE-2 will change the plan accordingly.

As shown in Figure 9 the new plan contains four main clusters of nodes; the first two (i.e., left-most) support the Defensive Air Superiority objective while the last two support the Offensive Air Superiority objective. The first cluster contains preemptive strikes against potential threats to Blue COGs; the second cluster establishes CAPs to defend identified Blue COGs. The third cluster establishes a breach of the enemy IADs; the fourth cluster extends air superiority over Red territory.

The second cluster of nodes from the left (i.e., that establish CAPs for Blue COGs) plays an important role in later stages of the demonstration. The topmost line in this cluster consists of

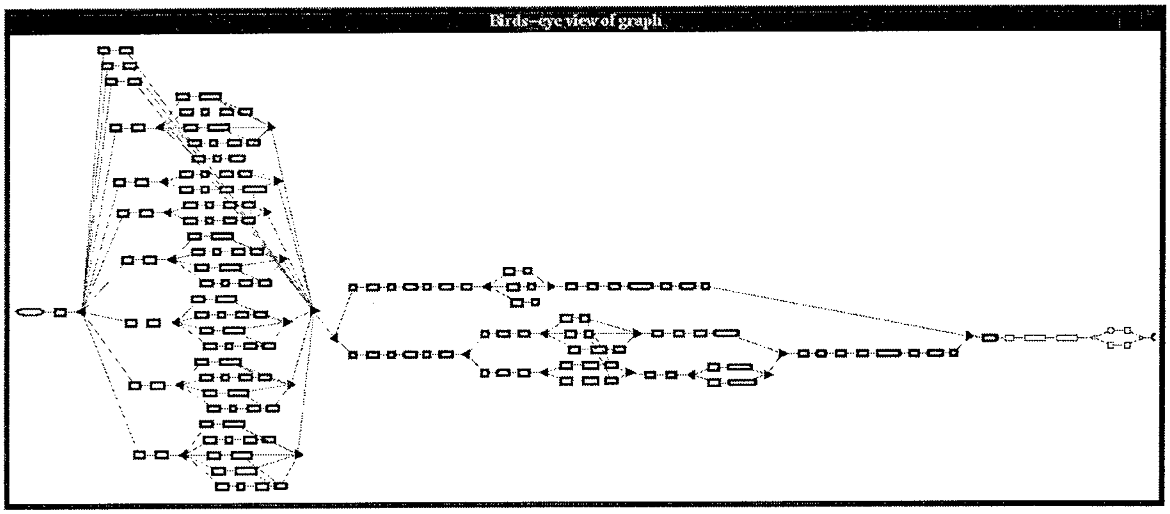

Figure 8: Birdseye View of the Initial Plan

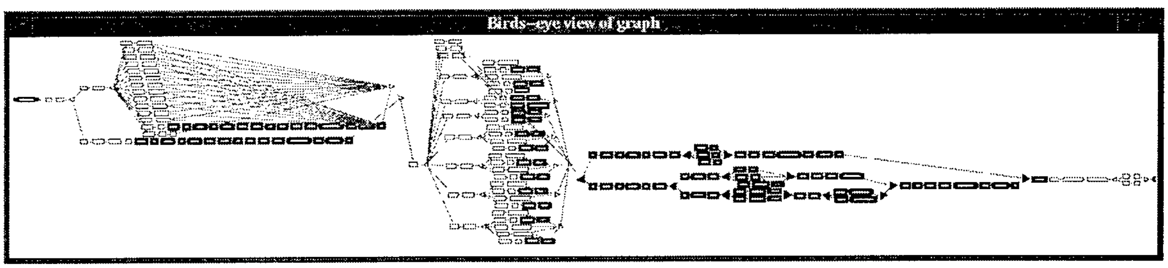

Figure 9: Birdseye View of the Modified Plan

two precondition nodes followed by a single action node. Examination of the third node shows the action `Unthreatened`, with effect `(Defend-COG Kennedy D+0 LOW)`. This set of nodes indicates that defense of the Kennedy requires no actions at this stage because it is believed that there are no immediate threats to the carrier. A later change in threat information will necessitate repair to this portion of the plan.

The next step in the demonstration is to install the plan for execution. From the `PLAN` menu, select the menu item `->PRS`. This will create an Act version of the plan and load it into the PlanMan agent. Additionally, it leads to the automatic generation of a set of monitors for testing relevant assumptions upon which the plan rests. Note that on sending the plan to PRS, the PlanMan I/O window displays the message that "Current Plan is now AIR-SUPERIORITY-A-V1-31-PLAN-1".

C.5.3 Plan Management

The next phase of the demonstration highlights CPEF's ability to perform situation monitoring with appropriate reactivity.

First, show the monitors that were created by the Planner in the last section. These monitors can be viewed by selecting the Act Editor system (through the `APPLICATION` menu) in the Executor. Within the Act Editor, select the `GRAPH` menu, and then the menu item `SELECT`. From the presented list of files, select the file `monitors-AIR-SUPERIORITY-A-17-PLAN-1.graph`.[9]

Next, select the `Act` menu, followed by the Select menu item. Select the third Monitor from the pop-up window that is labeled `(THREAT-AXIS)`. This monitor tracks `THREAT-AXIS` information that would impact the `Kennedy`, indicating that the plan should be repaired if new threats arise.

Now, switch to `PRS` from the `APPLICATION` menu. Then, switch to the `EXECUTION` menu within PRS. Evaluate `intel-update` in the LISP Listener. The value

```
(threat-axis sea-strike-threat Frans-Kaisiepo-Airbase Kennedy
             ec-pacific-coastal))
```

represents a change in intelligence information regarding possible threats to the Kennedy.

Send this update to PlanMan by evaluating the expression

```
(update-info intel-update)
```

The monitor examined above responds by requesting that the current plan be modified to reflect the new information. The PlanMan agent sends a request to the Plan Repair agent to oversee the plan repair process. The Plan Repair agent then invokes the Planner to make appropriate modifications to the plan.

A window for tracking the repair process will appear (see Figure 10). The repair process involves identifying affected portions of the plan for the updated world model, and then making modifications as appropriate. In this particular case, the planner recognizes that an additional PD-CAP mission is necessary to protect the Kennedy, and inserts it into the plan. Note that the repair process will take approximately a minute to complete.

The new plan is automatically sent to the PlanMan agent. Note from the PlanMan trace window that the new plan has now been installed.

C.5.4 Reactive Plan Tracking

The next phase of the demonstration is the simulated execution and tracking of the plan. Select the `Post Goal` menu item from the `Execution` menu, and enter the goal:

```
(EXECUTE-CURRENT-PLAN (AS PHASE-I D+0))
```

```
SIPE-2 Plan Update Trace (1)

Node to be copied and inserted:
GOAL:  P2379 (DEFEND-THREAT-AXIS LOCRIBLA STRIKE-THREAT DALTERIA-AIRBASE D+0 DAYTIME LOW) Phantomizes
Future precondition/phantom <PRECONDITION P4465> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS PLEONAUL-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2383 (DEFEND-THREAT-AXIS LOCRIBLA STRIKE-THREAT PLEONAUL-AIRBASE D+0 DAYTIME LOW) Phantomizes
Future precondition/phantom <PRECONDITION P4473> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS DALTERIA-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2388 (DEFEND-THREAT-AXIS TULJETI STRIKE-THREAT DALTERIA-AIRBASE D+0 DAYTIME LOW) Phantomizes:
Future precondition/phantom <PRECONDITION P4479> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS PLEONAUL-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2392 (DEFEND-THREAT-AXIS TULJETI STRIKE-THREAT PLEONAUL-AIRBASE D+0 DAYTIME LOW) Phantomizes:
Future precondition/phantom <PRECONDITION P4487> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS DALTERIA-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2397 (DEFEND-THREAT-AXIS KEFSUNU STRIKE-THREAT DALTERIA-AIRBASE D+0 DAYTIME LOW) Phantomizes
Future precondition/phantom <PRECONDITION P4493> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS PLEONAUL-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2401 (DEFEND-THREAT-AXIS KEFSUNU STRIKE-THREAT PLEONAUL-AIRBASE D+0 DAYTIME LOW) Phantomizes
Future precondition/phantom <PRECONDITION P4501> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS DALTERIA-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2406 (DEFEND-THREAT-AXIS GAAN STRIKE-THREAT DALTERIA-AIRBASE D+0 DAYTIME LOW) Phantomizes: P
Future precondition/phantom <PRECONDITION P4509> failed:
  (LEVEL> STRIKE-THREAT AIR-OPERATIONS PLEONAUL-AIRBASE D+0 DAYTIME 40);

Node to be copied and inserted:
GOAL:  P2412 (DEFEND-THREAT-AXIS GAAN STRIKE-THREAT PLEONAUL-AIRBASE D+0 DAYTIME LOW) Phantomizes: P

Adding new generated goal: GOAL:  P4769 (DEFEND-THREAT-AXIS KENNEDY SEA-STRIKE-THREAT FRANS-KAISIEPO
Replanning.
Planning level 1
Search has answer before critics, 1 alternative
Planning level 2
Search has answer before critics, 4 alternatives
Planning level 3
Search has answer before critics, 0 alternatives
Planning level 4
final plan produced: applying critics
recursion succeeded
recursion succeeded
SIPE-2 solved problem.
```

Figure 10: SIPE-2 Plan Repair Window

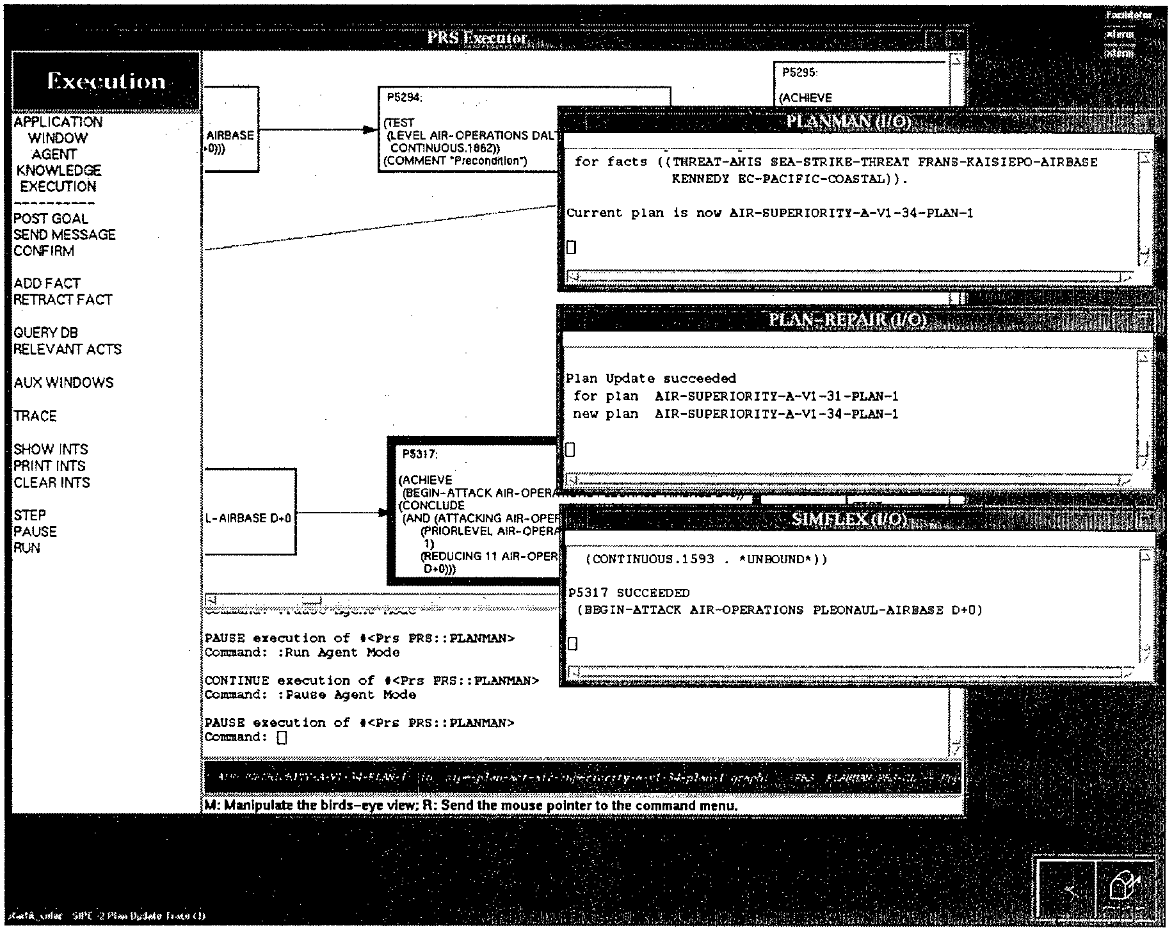

Figure 11: Tracking the Execution of the Plan

The graphical display will begin tracking the execution of the current plan and the trace window for SIMFLEX will display the simulation information being sent to PlanMan (see Figure 11).

Next, the demonstration shows how CPEF responds to various runtime events. First, evaluate `downed-report1` in the LISP Listener, to produce the value

```
(downed-pilot f-14d yuma-sector))
```

Send this message to PlanMan by executing the functional call

```
(update-info downed-report1)
```

PlanMan responds to this new information through another Monitor that has been defined (manually) within the system. Its response is to initiate a Search and Rescue mission based on a predefined standard operating procedure. Launching this plan results in the tracking and simulation of two plans simultaneously.

[9] If the demonstration is run multiple times, the exact name of the monitors file will vary slightly. However, it will always be of the form `monitors-PLANNAME.graph`.

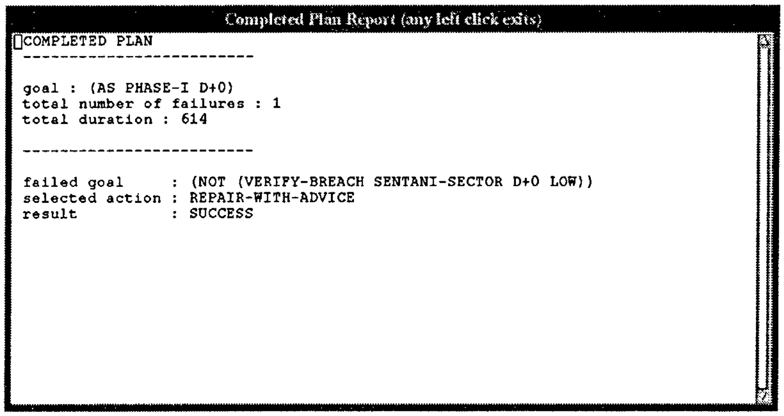

Figure 12: Completed Plan Report

Upon completion of the Search and Rescue mission, the demonstration proceeds to show how CPEF supports runtime plan repair. To initiate this phase, evaluate

```
(fail-task breach-wff)
```

in the LISP Listener. Doing so informs the system that the attempt to establish a breach in Sentani Sector was unsuccessful. Runtime plan repair will be invoked, but only after the node that validates the success of the breach has been reached (this node is purple in the Birdseye view). Reaching this point may require several minutes.

Plan repair is instigated by PlanMan issuing a request to the Plan Repair agent, which in turn manages the interactions with the Planner. The Plan Repair gives several options for dealing with the plan failure: ignore the problem, automatically repair the problem, or let the user advise the repair process. Choose `REPAIR-WITH-ADVICE` and the Planner will display the current advice. Select `:BREACH-AT-TWO-PLACES` and `:INGRESS-2-AT-ROCKATOON`.

Plan repair will take several minutes to complete, at which time the modified plan will be automatically loaded into PlanMan. A SIPE-2 window will appear that tracks the progress of the plan repair process.

Upon receipt of the modified plan, PlanMan performs a synchronization between the new plan and its current state of execution for the old plan. After performing the synchronization, PlanMan continues with its tracking process for the revised plan. When execution of the plan completes, a *plan report* window (Figure 12) will appear that summarizes key characteristics of the execution: execution duration, number and type of failures, and responses to detected failures.

C.6 Notes to the Demonstrator

The following notes are designed to help with recovery from problems that may arise during a demonstration.

- When entering an expression into the LISP Listener, typos may cause hard errors within CPEF (for example, when an unknown LISP function is evaluated). To recover, go to the LISP executable from which CPEF was launched. An error message similar to the following will be displayed, along with a variety of restart options:

```
Error: attempt to call 'UPDATE-INF' which is an undefined function
[condition type: UNDEFINED-FUNCTION]

Restart actions (select using :continue):
 0: Try calling UPDATE-INF again.
 1: Return a value instead of calling UPDATE-INF.
 2: Try calling a function other than UPDATE-INF.
 3: Setf the symbol-function of UPDATE-INF and call it again.
 4: Return to PRS:  Procedural Reasoning System command level
 5: PRS:  Procedural Reasoning System top level
 6: Exit PRS:  Procedural Reasoning System
```

  In general, selecting the action labeled `Return to PRS` (or `Return to SIPE`) is the best action to take, and should enable the system to recover completely from the original error. To select an option, enter `:cont <number>` and then press Return, where `<number>` is the identifier for the restart action to be executed (*e.g.*, here, `:cont 4`).

- If the following message appears at any time, right-clicking in the LISP Listener window will return you to the command prompt.

```
    The input "" is not a complete LISP expression.
    Please edit your input.
```

- A typing mistake in a goal expression or fact entered for a PRS agent will not cause hard errors. However, such mistakes will result in expected behavior failing to occur. If anticipated activity fails to occur when a goal or fact is sent to a PRS agent, try repeating the entry process.

- Should garbage collection occur frequently during demonstration, the amount of swap space allocation on the demonstration machine is inadequate. Ask your local systems administrator to increase the swap space for future demonstrations.

- When running under the Common Desktop Environment, changing workspaces may lead to the iconification of various CPEF windows. Thus, upon re-entering CPEF, any iconified windows will need to be manually re-opened.

D Glossary of Acronyms and Abbreviations

ACP	Air Campaign Planning
AI	Artificial Intelligence
AIM	Advanced ISR Management
AP	Advisable Planner
ARPI	ARPA/Rome Laboratory Planning Initiative
ATO	Air Tasking Order
BARCAPS	barrier CAPs
BE number	basic equipment number
C2	command and control
C3	command, control, and communication
CAP	Combat Air Patrol
CMU	Carnegie Mellon University
COA	course of action
COG	center of gravity
CPEF	Continuous Planning and Execution Framework
EW/GCI	Early Warning / Ground Control Intercept
DARPA	Defense Advanced Research Projects Agency
DAS	defensive air superiority
HTN	hierarchical task network
IADS	Integrated Air Defense System
IFD	Integrated Feasibility Demonstration
ISR	intelligence, surveillance, and reconnaissance
JFACC	Joint Forces Air Component Commander
JOM	JFACC Object Model
JPR	JFACC Plan Representation
KB	knowledge base
LOC	lines of communication
MPA	Multiagent Planning Architecture
NASA	National Aeronautics and Space Administration
OCA	offensive counterair
OAS	offensive air superiority
PD	point defense
PRS	Procedural Reasoning System
SAM	surface-to-air missile
SDA	Strategy Development Assistant
SIMFLEX	Simulated Flexible Execution
SOF	Special Operations Forces
SUO	Small Unit Operations
SWIM	Smart Workflow for ISR Management
TIE	Technology Integration Experiment
TIED	Technology, Integration, Evaluation, Demonstration
USAF	United States Air Force
USC/ISI	University of Southern California/Information Sciences Institute
USN	United States Navy

65

DISTRIBUTION LIST

addresses	number of copies
JOHN SPINA AFRL/IFTD 525 BROOKS ROAD ROME, NY 13441-4505	5
SRI INTERNATIONAL 333 RAVENSWOOD AVENUE MENLO PARK, CA 94025-3493	5
AFRL/IFOIL 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	1
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P.STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

AFRL/HESC-TDC 2698 G STREET, BLDG 190 WRIGHT-PATTERSON AFB OH 45433-7604	1
ATTN: SMDC IM PL US ARMY SPACE & MISSILE DEF CMD P.O. BOX 1500 HUNTSVILLE AL 35807-3801	1
TECHNICAL LIBRARY D0274(PL-TS) SPAWARSYSCEN 53560 HULL ST. SAN DIEGO CA 92152-5001	1
COMMANDER, CODE 4TL000D TECHNICAL LIBRARY, NAWC-WD 1 ADMINISTRATION CIRCLE CHINA LAKE CA 93555-6100	1
CDR, US ARMY AVIATION & MISSILE CMD REDSTONE SCIENTIFIC INFORMATION CTR ATTN: AMSAM-RD-OB-R, (DOCUMENTS) REDSTONE ARSENAL AL 35898-5000	2
REPORT LIBRARY MS P364 LOS ALAMOS NATIONAL LABORATORY LOS ALAMOS NM 87545	1
ATTN: DEBORAH HART AVIATION BRANCH SVC 122.10 FOB10A, RM 931 800 INDEPENDENCE AVE, SW WASHINGTON DC 20591	1
AFIWC/MSY 102 HALL BLVD, STE 315 SAN ANTONIO TX 78243-7016	1
ATTN: KAROLA M. YOURISON SOFTWARE ENGINEERING INSTITUTE 4500 FIFTH AVENUE PITTSBURGH PA 15213	1

USAF/AIR FORCE RESEARCH LABORATORY 1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB MA 01731-3004

ATTN: EILEEN LADUKE/D460 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

OUSD(P)/DTSA/DUTD 1
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 MARIE A. BIENKOWSKI 1
SRI INTERNATIONAL
333 RAVENSWOOD AVE/EK 337
MENLO PRK CA 94025

DR MARK S. BODDY 1
HONEYWELL SYSTEMS & RSCH CENTER
3660 TECHNOLOGY DRIVE
MINNEAPOLIS MN 55418

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

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

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
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE, OR 97403

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

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

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

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

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
NEW 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 22203-1714

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

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

DR MATTHEW L. GINSBERG 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE OR 97403

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

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

DR CARLA GOMES 1
AFRL/IFTB
525 BROOKS RD
ROME NY 13441-4505

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
TSX CORPORATION
4301 N FAIRFAX DRIVE, SUITE 301
ARLINGTON VA 22203

DR TOM GARVEY 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

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

OFFICE OF THE CHIEF OF NAVAL RSCH 1
CODE 311
800 N. QUINCY STREET
ARLINGTON VA 22217

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

DR CHRISTOPHER OWENS 1
GTE
10 MOULTON ST
CAMBRIDGE MA 02138

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 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

MR.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

GINNY ALBERT 1
LOGICON ITG
2100 WASHINGTON BLVD
ARLINGTON VA 22204

ADAM PEASE 1
TECKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

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

DR. STEPHEN E. CROSS, DIRECTOR 1
SOFTWARE ENGINEERING INSTITUTE
CARNEGIE MELLON UNIVERSITY
4500 FIFTH AVE 15213
PITTSBURGH PA 15213

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

NSA/CSS 1
K1
FT MEADE MD 20755-6000

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

THE MITRE CORPORATION 1
D460
202 BURLINGTON ROAD
BEDFORD MA 01732

DR. DAVID ETHERINGTON 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE, OR 97403

DR. MAREK RUSINKIEWICZ 1
MICROELECTRONCS & COMPUTER TECH
3500 WEST BALCONES CENTER DRIVE
AUSTIN, TX 78759-6509

MAJOR DOUGLAS DYER/ISO 1
DEFENSE ADVANCED PROJECT AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

DR. STEVE LITTLE 1
MAYA DESIGN GROUP
2100 WHARTON STREET S&E 702
PITTSBURGH, PA 15203-1944

NEAL GLASSMAN 1
AFOSR
110 DUNCAN AVENUE
BOLLING AFB, WASHINGTON, D.C.
20332

AFRL/IFT 1
525 BROOKS ROAD
ROME, NY 13441-4505

AFRL/IFTM 1
525 BROOKS ROAD
ROME, NY 13441-4505

DR. CHARLES L. MOREFIELD 1
ALPHATECH, INC.
2101 WILSON BLVD, SUITE 402
ARLINGTON VA 22201

MR. GARRY W. BARRINGER 1
TECHNICAL DIRECTOR
AEROSPACE CZ ISR CENTER
LANGLEY AFB VA 23665

DR. JAMES HENDLER 1
DEFENSE ADVANCED PROJECT AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

☆U.S. GOVERNMENT PRINTING OFFICE: 2000-510-079-10014

MISSION OF AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.