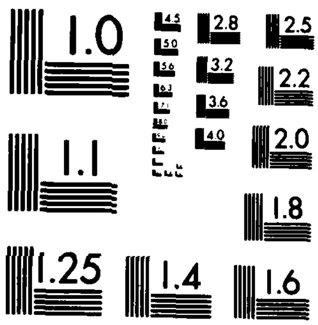
AD-A151 869 AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING D L KNODE DEC 84 1/2

UNCLASSIFIED AFIT/GCS/ENG/84D-15 F/G 6/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN

THESIS

David L. Knode
Captain, USAF

AFIT/GCS/ENG/84D-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

85 03 13 075

AFIT/GCS/ENG/84

AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN

THESIS

David L. Knode
Captain, USAF

AFIT/GCS/ENG/84D-15

AFIT/GCS/ENG/84D-15

AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Science

David L. Knode, B.S.

Captain, USAF

December 1984

Preface

The purpose of this research was to determine if Dr. Robert Wilensky's theory of planning is appropriate for a pilot aid in the aircraft inflight emergency domain.

The loss of the lives of several of my friends in aircraft accidents motivates my interest in this project.

I thank my advisor, Captain Stephen E. Cross of the Air Force Institute of Technology, Artificial Intelligence Laboratory, for his motivation, patience, and guidance. I thank my reader, Captain Robert W. Milne, also from the Artificial Intelligence Laboratory, for his philosophy and technical critique. I thank my sponsor, the Avionics Laboratory/System Avionics Division of the Air Force Wright Aeronautical Laboratory, represented by Capt Gregg Gunsch, Avionics Design Engineer, for their advice and "real-world" updates. I also thank Major Donald M. Drollinger and Captain Jes Barbera for their flight domain expertise and the AFIT library and teaching staff for their support and contributions.

I am deeply indebted to my wife, Keiko, for her selfless support and the gift of a wonderful newborn son. I humbly thank the good Lord for this opportunity to grow intellectually and spiritually.

David L. Knode

Table of Contents

List of Figures

Abstract

A planner must understand its domain and be able to effectively reason about interacting goals competing for satisfaction in its environment. Current artificial intelligence (AI) planning structures are inadequate for expert and commonsense reasoning in the dynamic aircraft inflight emergency domain. These current planners are inadaquate because they are not designed to manipulate multiple goals in an unpredictable environment, nor are they equipped to simulate dynamic, time dependent processes. Conflicting goals, i.e., when the realization of one goal interferes with the realization of another goal, poses particularly frustrating problems for typical planners.

This research discusses a new planning approach for the inflight emergency domain based on Wilensky's planning theory for the everyday activities domain. Like the everyday activities domain, the flight domain requires a large pool of world and common sense information. It also requires flight domain information and knowledge of the goals and plans of its pilots and other aircrew members.

The planner for an intelligent pilot aid (PIPA) divides planning into four activities, 1) goal detection, 2) plan proposition, 3) plan projection, and 4) execution; composed into four components with similar names. Goals are associated with the observations which trigger them, and

plans are associated with the goals to which they apply. Proposed plars are simulated in a hypothetical model of current states and are watched for weaknesses, overlap, or conflict. Overlapping plan steps are appropriately combined, while conflicts direct the PIPA to either prospose and test new plans or have the PIPA attempt to change the circumstances surrounding the conflict. When a conflict cannot be resolved, the less important goal is abandoned.

Implementation of the PIPA was partially completed, but more research in the areas of simulation and model-based reasoning is required. Qualitative reasoning and common sense algorithm (CSA) representation are proposed as possible solutions toward this end.

AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN

I. Introduction

Background

Pilots make mistakes which result in the loss of time, material, and life. In the period 1979 through 1983, the United States Air Force lost 204 aircraft and 305 lives due to operator error (Air Force Safety Center, 1984). Civilian losses were also significant during this period. Historical solutions or fixes for these operator induced accidents include standardized procedures for communication and navigation; improved equipment performance, feedback, and reliability; and intensive, realistic simulator training. Certainly these measures have helped reduce the number of fatal accidents; unfortunately, too many still occur. These continued losses have prompted research in the area which involves the design of "intelligent" decision-aids in the flight management domain. These aids will attempt to reduce pilot task saturation and to increase the emergency situation survival probability.

Three levels of decision-aids have been identified in this research of the feasibility of an "intelligent cockpit" (Hammer, 1983:2). Most work and successful products have been at the lowest aid level which includes computerized warnings, real-time calculations, and display control (Heads Up Display, e.g.) (Wiener, 1980). Much less work and success occurs at the second and measurably more difficult level, which requires the decision aid, or pilot aid (PA), to be capable of monitoring and inference. At level two the PA observes the pilot's actions, determines what "plan" he is using, and then "follows" along to insure that the pilot executes the "plan" correctly. For example, if the pilot reduces the power, begins a descent to a lower altitude, and lowers the landing gear, the PA should infer that the pilot intends to land the aircraft. The PA then matches the pilot's actions against those outlined on its stored "landing procedure plan." If a match between the pilot's actions and a stored plan is found, the PA continues to monitor the pilot's actions. If no match, the PA tells the pilot to reconfirm his actions. These concepts of monitoring and inference are being studied by several Artificial Intelligence (AI) reseachers, especially in the area of "expert systems."

The third and most difficult level is the level at which the intelligent pilot aid (IPA) can advise the pilot of, or compensate for, pilot error. Advice could be in the form of emergency procedure planning before an error,

Pilot to IPA, "We've just lost number two engine.
What should we do?"

IPA to Pilot, "Feather number two.
Declare an emergency.

Head for Right-here airport."

or after an error,

IPA to Pilot, "Sir, you've just shut down the wrong engine!"

In the landing example above, if the pilot has forgotten to position the flaps in the landing position, the intelligent pilot aid would advise or question the pilot about the flaps. The pilot would then correct the configuration, or he would explain to the IPA why he's flying with the seemingly incorrect configuation. Some systems could be designed to allow the IPA to initiate the correction itself; however, there will be great pilot resistance to this in the areas of flight controls, power control, and weapons delivery (author's experience). While much progress has been made in building and under_tanding pilot aids at level one, machines able to perform at levels two and three do not yet exist.

Despite the advances in flight management aids, pilots continue to make fatal errors. Many aircraft today are equipped with inflight data recorders which record engine settings, aircraft configuration, cockpit voices, and other information helpful in determining the cause of the accident. Unfortunately, this method of error detection is applied after the accident, too late to negate any detrimental effects of the mistake. Critical errors must be detected within seconds or minutes after occurrence, or ideally, prevented from occurring at all. Artificial Intelligence applications are being investigated, but at this time no machine system is able to reason and understand in this dynamic flight environment, let alone give assistance at these higher levels to the well-trained professional pilot. These requirements pose several difficult problems for the builders of IPA systems.

Problem

The problem is that the traditional artificial intelligence planning approaches are inadequate for handling the multiple goal conflicts encountered in the dynamic inflight emergency domain. These approaches are insufficient primarily because they have not been designed to operate in this type of environment. These approaches generally deal with a single, a priori specified goal in a difficult, though static, domain. Planning a chess move or making a medical diagnosis are representative examples. The inflight emergency domain requires a planner capable of autonomous goal detection and the ability to handle multiple goals interacting in a dynamic environment. Requiring a large database of world and flight knowledge, the planner has to fuse together the plans for multiple goals, a relatively easy task for most humans to perform. For example:

(1) On his way home from work, Joe stopped at a gas station and filled his car's gas tank.

Joe's actions were really the combination of his "go home plan" and his "get gas plan." Joe could have driven home, and then driven to a gas station, but he devised the more efficient plan to get the gas on his way home, saving an extra trip. In this case, the goal interaction involved combining the plans into a single, optimum plan.

An example of goal conflict introduces more complexity into the planning process:

(2) The school nurse announced that flu shots would be given the next day. Little Tim wanted to be healthy, but he also wanted to avoid pain. Tim got his shot the next day.

Though Tim knew that getting the shot would probably cause some pain, he reasoned that the consequences of not getting the inoculation, possibly catching the flu, would be more painful than the momentary pain of the injection. He had to project into the future the possible courses of action, apply his knowledge, and then choose a plan which would best satisfy his conflicting goals, abandoning one if necessary.

The inflight emergency domain involves not only rich interactions of goals and plans, but an environment which itself can be rapidly changing. Weather, terrain, enemy actions, airspeed, altitude and aircraft configuration and power all can vary with time, and so need to be monitored for their direct influence on the planning procedures.

Several other key problems in planning are identified by Hayes-Roth (1983:84), but this research will focus on the problems mainly associated with goal conflicts.

Scope

The scope of this research is to determine whether or not Wilensky's (1983a) planning theory is adequate to successfully handle the problem of multiple goal conflicts characteristic in the inflight emergency domain. This thesis describes a conceptual design based on Willensky's theory of planning in the mundane activities domain for an IPA "planner" in the C-130 inflight emergency domain. A partial implementation of the design using current AI programming techniques is used to analyze portions of the planner with emergency situation examples rich with goal interactions. Design and implementation discussions address some of the relavent issues of current AI research. An analysis of the planning approach and implementation is provided. Recommendations for future research are also provided.

Treatment of the design and implementation of sensor fusion, communication interfaces, data acquisition and integration, and whole system management and control for an IPA is not discussed.

Assumptions

Since this planner would make up only a part of the total IPA, it is assumed that the other parts of a fully functional IPA exists. These other parts or modules could include a "navigation and orientation" package, an "aircraft subsystems' operations" package, a "weight and balance" monitor, an "offensive weapons director," and a "defensive weapons director." Each module is linked to and controlled by a "command module" responsible to the pilot and aircraft.

It is assumed that sensory knowledge available to the pilot is simultaneously available to the IPA. The IPA can "listen" to radio calls and interphone conversations, it can "see" the flight instruments, performance guages, and outside references such as mountains, runways, and thunderstorms. The IPA can also "feel" the sensations of yaw, thrust, and gravity as they affect the aircraft and crew. A natural language communication interface allows the IPA and pilot to "talk" to each other using a limited, flight domain vocabulary.

These assumptions are necessary because in the inflight emergency domain, time is of the essence. Pilots use the expression "He's ahead of the aircraft" to mean that the pilot is aware of what's going on and anticipates the next situations. "Behind the aircraft" indicates the pilot is slow to react, missing cues, and displaying subnormal performance. For a planner to be "ahead of the aircraft," it must recieve real-time information. Like the pilot, the IPA planner must continually review and update plans based on new information.

Frequently the cause of an aircraft accident must be speculated because the incriminating evidence is destroyed in the crash and subsequent fire. It is assumed that the cause of some accidents is the pilot's lack of a memorized plan or lack of sufficient time to decide on a plan of action before the accident occurrs.

Summary of Current Knowledge

Although very little artificial intelligence research has been done specifically for third level IPA systems (Rouse, 1982), much work has been done in the area of planning and plan construction, usually in domains involving a single goal. But the inflight emergency domain involves multiple goals with interacting plans, plans which taken by themselves are not difficult to construct. The difficulty involves attempting to merge several plans together to accomplish the multiple goals. "The problem of interacting plans has long been recognized," (Wilensky, 1983a:14) and studies by Sussman (1975), Sacerdoti (1977), Tate (1975), and Warren (1974) all outline approaches for handling these interactions. Wilensky contends that these approaches "fail to sufficiently emphasize the complexity and significance of the interactions in the planning process" and that the method of handling these interactions should "be moved from its secondary status to the primary framework around which the planner is designed" (Wilensky, 1983a:14). By failing to emphasize the complexity, they also fail to intelligently handle goal overlap and especially goal conflict.

By building the planning structure around the notion of goal interactions, the power of the total planner itself is available to help solve problems instead of needing to rely on critics or passing the planning problem off to another planning plane. This structure allows to the planner to always know why it has done what it has done. If one has an expert solve his problem, has he increased his understanding? As much as if he'd done it on his own?

A close look at these traditional planning approaches reveals why they are insufficient for the inflight emergency domain. An early approach, non-hierarchical planning, suffered from its inability to distinguish the important problem-solving elements of a plan from the unimportant details. A planner, such as STRIPS (Fikes, 1971) or GPS (Newell and Simon, 1972) using the problem-solving method of means-end analysis, examines its current state and compares it to the desired or goal state (Barr, 1981:113,129). It then selects and applies an operator to reduce the difference or distance between states, and iterates until the goal state is met or until a precondition is not satisfied. If the precondition failed, the problem-solver must "backtrack" until it finds another option or path to try. Backtracking in a nonhierarchical structure becomes extremely time consuming as the size of the problem domain increases. Not only will unnecessary details be processed, but following a backtrack, the problem-solver may need to reprocess some of the details again. Because it's expensive, backtracking is usually minimized (Cohen, 1982:526).

The solution to the problem of premature detailed planning was the hierarchical planning approach. This approach divides a plan into a hierarchy of abstraction levels and then outlines or sketches a basic plan at the highest level, using just the most essential parts of the plan. It then refines the subparts of the plan until a final, detailed plan exists. This delaying of the detailed planning saves time because less backtracking occurs. In short, the hierarchical planner employs the tactic of delaying the minute planning details until after the main portions of the plan have been decided.

Two of the finer examples of hierarchical planners include NOAH by Sacerdoti (1977) and MOLGEN by Stefik (1980). Nets Of Action Hierarchies uses a "procedural net" of domain knowledge and plan knowledge. Procedures called "critics" contain knowledge about how to detect and fix certain defects in plans. A problem with this approach is that only the power of that critic focusses on the problem, rather than the whole planning system. NOAH's critics used for eliminating preconditions do not give enough complex consideration to the preconditions and the relations to the plan operations. For example, in Sacerdoti's paint the ladder and ceiling example, a critic removes one of the redundant preconditons "get paint" without considering how much paint, what colors, etc. Basically then, a critic doesn't necessarily know why it is doing what it does, just that it is supposed to do it's job when the situation warrants action without regard to the consequences it may have on the goal.

In MOLGEN, planning control is divided among three layers or spaces; planning, design, and strategy. Interactions between subproblems are represented as structures called "constraints" and are used to help quide the planning activity (Cohen, 1982:552). The problem of the notion of constraints is that they can only block plans, rather than propose new ones (Wilensky, 1983a:36). However, this planning approach is suitable for and is being used in several complex domains, including a route planner for air launched cruise missiles (Millar, 1984).

Wilkins (1982) has designed and implemented a domain independent planner system called SIPE (System for Interactive Planning and Execution monitoring) which uses hierarchical planning and parallel actions in its approach. He claims several significant improvements over systems like NOAH and MOLGEN, but admits that "sophisticated reasoning about time and modelling of dynamic processes are not possible within our present framework" (Wilkins, 1982:5). The inflight emergency domain certainly involves time and dynamic processes.

Another planning approach is the script-based planning approach. Many flight procedures and emergency flight procedures characterize the notion of scripts as described by Shank (1977). A script "is a predetermined, sterotyped sequence of actions that defines a well-known situation," (Schank, 1977). For example, an experienced pilot merely references his "Before-Landing" script, his "Go-Around" script, his "Landing" script, etc, to accomplish his short term, frequently occurring goals. When encountering a situation with no known script, the pilot, using his powers of reasoning and understanding, references his knowledge of the various factors of his situation and creates a new script for solving the problem or achieving the goal at hand. The problem with this approach is that it is unable to combine several plans into one to achieve the multiple goals.

Another inadequate planning approach for this domain is the "opportunistic planning model" proposed by Hayes-Roth (Hayes-Roth, 1979, 1980). Planning is viewed as the cooperative efforts of numerous specialists who post their tentative solutions on a "blackboard" for all to see. New decisions can be made by reference to this blackboard, which is basically a data-structure divided into planes of five different decision categories. These categories include (a) metaplan decisions or general approach; (b) plan decisions or actions to take; (c) plan abstraction decisions or desirable actions; (d) world knowledge decisions; and (e) the planning process itself or executive decisions. These planes are further divided into levels of abstraction (Hayes-Roth, 1980:v).

The problem with this approach is that this rich structure "is susceptible to these (subgoal) interactions," and "is more likely to need to rewrite parts of its plan or change its goals than is a hierarchical planner" (Cohen, 1982:24). The opportunistic approach tends to be one of "planning as required" as opposed to the imposed "plan ahead" requirement of the inflight emergency domain.

Some of the current research (Schira, 1984; Anderson, 1984) in the area of expert system pilot aids involves the use of rule based production systems. There are a few advantages with this approach. New rules can be easily added and current rules can be modified or deleted without difficult changes to the control structures. The "if-then" concept is logical and easy to understand by its users. However, there are some major shortcomings with this approach also. It will be extremely difficult to encode in rule form the rich combinations of possible emergency situations which require pilot action. Not only would the number of rules be excessively large, the exceptions to the rules are nearly as numerous as the rules themselves.

Rules are not synonymous with understanding. One can know a rule about something, but have no understanding why the rule is true nor be able to reason about its consequences. Also, a rule based approach will provide little assistance outside its specific circumstances.

Approach

The approach followed for this research is typical of other conceptual design studies. A literature search on AI planning approaches provided the references for the research. After inadequecies of traditional planning approaches were identified, the next step involved the study of Wilensky's planning theory, the proposed solution to these traditional shortcomings. Since no critique or review of his works by other AI researchers had yet been published, Wilensky's own writings were the source for this part of the study. Wilensky's view of meta-planning, which differs form Davis' (1977) and Barr's (1977) view, is presented.

The third step was to describe and characterize the C-130 inflight domain and inflight emergency domain. The source of this information includes the personal experience of this author, a USAF C-130 Aircraft Commander with more than 2000 military flying hours; a USAF pilot currently serving as a wing safety officer and a former standards and evaluations pilot; and a third USAF pilot serving as the chief of standards and evaluation for an overseas HC-130 squadron. Other references include the applicable flight regulations, manuals, and checklists.

The conceptual design step followed the C-130 inflight emergency domain description. The design outlines the requirements of an inflight emergency planner, its components, and a discussion of its structure and function.

A partial implementation of the design was built, incorporating currently available AI programming techniques. A discussion of possible plan evaluation and simulation approaches is offered. The implementation was analyzed using examples from the inflight emergency domain.

Finally, an analysis was conducted (1) to determine whether or not Wilensky's theories on planning are appropriate for the inflight emergency domain; (2) to see if current AI programming techniques are complete enough to implement the planning model design; and (3) to discover any changes that might improve the planner design for this inflight emergency domain.

Materials and Equipment

The design was implemented in Franz LISP on a Digital Equipment Corporation VAX 11/780 computer in a multi-user environment.

II. WILENSKY'S PLANNING THEORY

Introduction

Wilensky's planning theory was motivated by his "study of the inference procedures required for natural language text understanding" (Wilensky, 1983:xi). He was frustrated in his attempt to build a text understander because none of the then current theories of plan structure could account for the richness of everyday situations. He reasoned that if these structures were unable to understand common sense situations, they would also be unable to "plan" in these same situations. So Wilensky began his research for the development of a planning structure which could account for the common sense, everyday activity environment.

His research is heavily influenced by the work of Schank and Abelson (1977) in the area of scripts, plans and goals, by McDermont (1977) in the area of planning conception, by Sacerdoti (1977) in the area of goal interaction, and by Carbonell (1979) on goal competition.

Principles of His Theory of Planning

Wilensky contends that his planning structure for the mundane, everyday activity domain is richer and probably more complex than other planners in domains of difficult tasks (e.g., chess, geology, genetics, etc.). It is not that the individual plans in commonplace activities are so complex, but that rather it is the immense number of goals and plans and their interactions which require the complex structure. Consider, for example, the goal of having a nice dinner with Peggy. The plans such as eating in a restaurant, driving somewhere, making a reservation, asking for a date, getting dressed, making light conversation, and deciding on a menu selection all have to be meshed together into a reasonable course of action. Other aspects of this domain contribute to the complexity of its planning structure. For example, what happens when after you arrive at the restaurant you notice that Peggy's football star boyfriend is at the restaurant with some of his teammates?

An effective planning structure in the domain of everyday activities requires a large database of "world knowledge." This world knowledge contains facts about cars, houses, animals, eating, shopping, etc. In order to use these facts in an intelligent manner, the planner must understand the commonsense relationships between these facts. For example, this dialog between the user and the planner's database suggests the subtle consequences of some misunderstood relationships:

USER: Do submarines have screendoors?
DATABASE: No.
USER: Why?
DATABASE: Screendoors are used to keep flies out of a house. Flies do not live underwater, so submarines do not use screendoors!

The point made above is that even though the first answer is correct, it is correct for the wrong common sense reason. The fact that screendoors will not keep out the water should be known or at least deduced by the database, else the value of it abilities are certainly limited.

Wilensky increases the planner's complexity by requiring it to be autonomous in detecting its own goals based on the situation it finds itself in, rather than simply having the goals handed to it. This capability is especially important for story comprehension and speech understanding. A security robot, as another example, may have the goals of patrolling the warehouse, alerting help when appropriate, keeping itself supplied with charged batteries, and staying out of the way of human workers. These goals would not normally all be active at the same time, requiring the robot to "know" about its goals and when each goal is appropriate.

The planning structure must have goals of its own, such as producing plans that are not wasteful. For example, buying gas on the way home is a more efficient and hence, more desireable plan than first going home and then driving back to the station for a fill-up. This type of knowledge about planning, that is, the goal of the planning process itself, is called the "meta-planning" knowledge. This presentation of its own goals and plans to itself underscores the priciples of Wilensky's planning theory.

That is, the same structure which detects goals and builds plans is the same structure which detects the planning goals and builds the planning plan.

After a plan is built, it is tested for success by a simulation mechanism called "projection." Generally a first-pass plan is flawed and needs to be refined or changed altogether. Projections can spot these problems in a proposed plan, and have the plan passed back for an improved one. The projection process continues even after a suitable plan is discovered, watching for any updates or modifications that may be required.

In short, Wilensky's notion of planning includes assessing a situation, determining what goals to pursue, finding or building plans to attain these goals, and executing these plans.

Meta-Planning

Wilensky expresses the knowledge about how to plan in the terms of a set of goals (meta-goals) for the planning process and a set of plans (meta-plans) to achieve these goals. The planner's meta-planning knowledge also includes meta-themes which specify when or under what circumstances the planner should have those particular meta-goals. These meta-themes guide the planner through its planning process.

Four meta-themes comprise this guidance package. They include:

1. DON'T WASTE RESOURCES.
2. ACHIEVE AS MANY GOALS AS POSSIBLE.
3. MAXIMIZE THE VALUE OF GOALS ACHIEVED.
4. AVOID IMPOSSIBLE GOALS.

The planner begins its planning task under the DON'T WASTE RESOURCES theme which encourages the production of efficient plans. For example, washing a load of clothes once per week is more efficient than washing two or three items everyday. Encountering a goal conflict summons the ACHIEVE AS MANY GOALS AS POSSIBLE theme which tries to resolve the conflict. Using a calorie-free sugar substitute is an example of a recurring plan for resolving the conflict between enjoying an otherwise fattening food and not gaining weight. If the conflict cannot be resolved, the theme MAXIMIZE THE VALUE OF GOALS ACHIEVED suggests forgetting the less valuable goals and concentrating on the more important ones. Goals with impossible plans are deleted by the AVOID IMPOSSIBLE GOALS meta-theme.

Requirements of Wilensky's Planner

Wilensky outlines seven requirements for a planner in the everyday activities domain (Wilensky, 1983:19,20).

1. Plans are associated in memory with the goals t ·hich they apply.

2. Plans that are associated with a particaular goal can retrieved from memory by specifying that goal.

3.The planner can project plausible, hypothetical futures from its knowledge of the present world together with its own tentative plans.

4. Goals can be inferred based on the situation in which a planner finds itself.

5. The planner must be capable of detecting the interactions between its goals.

6. These interactions must be taken into account in its subsequent planning processes.

7. In addition to generating and modifying plans for goals, the planner must be capable of evaluationg scenarios and of abandoning some goals in order to secure others.

Major Components of the Planner

Wilensky proposes a four component planner consisting of a "Goal Detector", a "Plan Proposer", a "Projector", and an "Executor." The structure is designed to allow a flowing effect or continuous flavor to the planning process.

The "Goal Detector" begins the planning process via its mechanism called the "Noticer" which recognizes something it was instructed to look for (a warning light, a great sale price, hunger, etc) and then passes this information to the rest of the "Goal Detector" which now simply finds the goal associated with that something. The "Goal Detector" then passes this newly detected goal to the "Plan Proposer." The "Plan Proposer" looks for any stored plan which may achieve this goal and passes it to the "Projector." Not limited to merely finding a canned plan, the Proposer can edit previously successful plans or devise entirely new ones applicable to the goal.

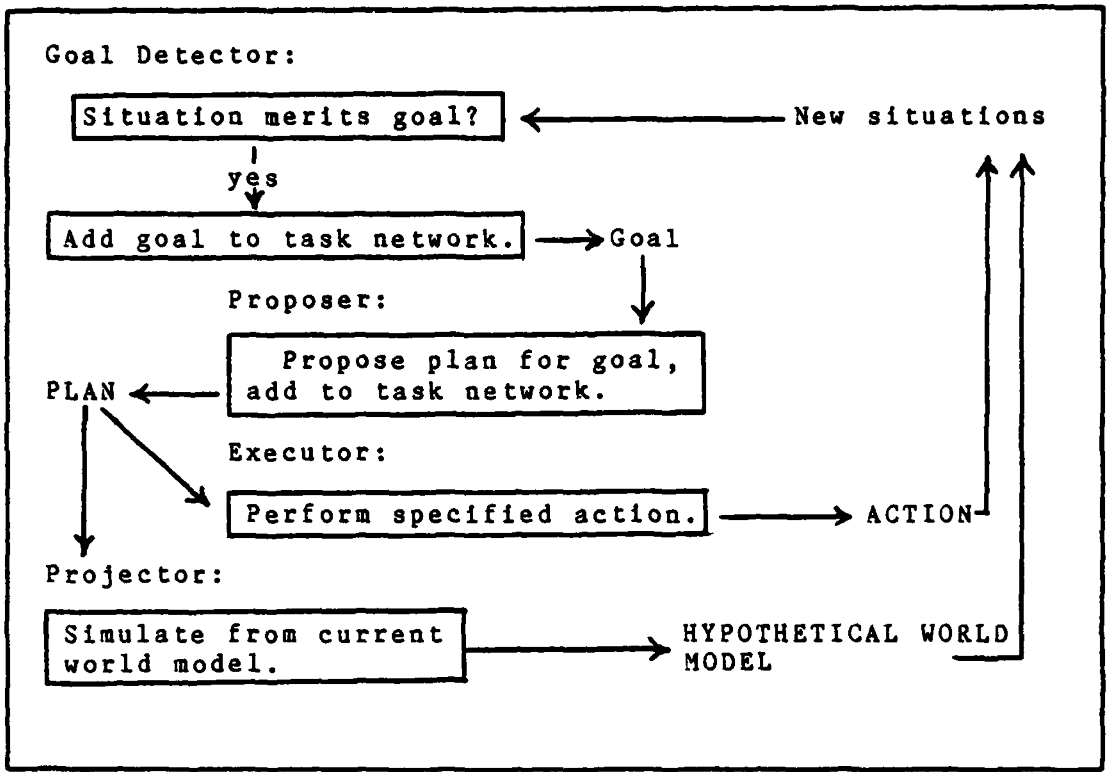
The "Projector" is the most powerful and complex component and has the task of responsibly managing the goal interactions. After it recieves a plan from the "Plan Proposer," the "Projector" tests this plan by simulating it on a model of the current "world state." If all the plan's preconditions are met and none of its steps conflict with other plans, it then gets passed to the "Executor."

During the Projector's simulations, the "Goal Detector" watches, looking for the same things it was instructed to look for in the "real" part of the planner. In other words, new goals can be detected and planned for in the simulation process itself. This nesting, iterative process requires a complex management structure.

If the proposed plan causes a conflict with another more important goal, or its required preconditions were not met, then it will be returned to the Proposer along with the reason for its failure. The Proposer can now use this information for selecting the next plan. This process will continue until either a good plan is found or all plans have been tried. If no successful plan can be found, the goal may have to be abandonned.

The "Executor" recieves the approved plan and begins the execution. The plan becomes final when it has been fully executed.

Figure 1 diagrams a design of the planner components, with the arrows indicating inputs and outputs. The Goal Detecter discovers a goal and adds it to the task network. The Plan Proposer adds a proposed plan for the new goal to



(Wilensky, 1983a:23)

Figure 1. The Components of the Planner

the plan network. The Projector simulates this plan in its hypothetical world model and, if successful, passes it to the Executor for action. Rejected plans get passed back to the Proposer for another one, if available.

Applications and Success

This planning representation has been implemented in two domains using PANDORA (Plan ANalysis with Dynamic Organization, Revision, and Application), a plan generation program (Wilensky, 1983a:151, Faletti, 1982). In the domain of mundane activities, PANDORA resolves simple goal conflicts such as going outside to get the newspaper while it is raining and staying dry. In the domain of the UNIX operating system, it acts as a consultant on the system itself. A novice user asks a question in a natural language format, and the "model" gives a reply. For example, the user types in "How do I delete a file?" and the response he gets is "Typing ´rm filename´ will remove the file with name filename from your current directory." The model treats the goal of the question as its own goal and then plans the solution. The goal conflict of trying to add a file to an already full directory was temporily met by mailing a copy of the file to yourself and then requesting more space from the operator.

Wilensky argues that "expert reasoning" is simply applying common sense reasoning to a more esoteric domain. The application of his planning model to the UNIX operating system domain strengthens this theory (Wilensky, 1983:151). This thesis analyzes his claim using a modification of his theory in the inflight emergency domain.

Pilot Aid Application

Chapter three discusses the C-130 inflight emergency domain and suggests that Wilenky´s planning theories may be suitable for building a planning aid for pilots.

III. THE C-130 INFLIGHT EMERGENCY DOMAIN

Introduction

The C-130 inflight emergency domain is similar to the domain of routine everyday activities modelled by Wilensky, with some important differences. This chapter describes the C-130 inflight emergency domain, identifies its similarities with the ordinary activities domain, and describes the important differences which exist between them. This chapter concludes with a pilot's planning approach for the safe termination of two inflight emergencies.

The Lockheed C-130 Hercules is a multi-mission, medium-range, four-engine turboprop, multi-crew aircraft. Minimum crew consists of a pilot, co-pilot, flight engineer, and load master, but may include one or more navigators, a radio operator, and rescue specialtists, depending on configuation and mission. The majority of the C-130s in the USAF inventory support the tactical airlift mission, the rapid transportation of cargo or personnel for delivery by parachute or by landing.

Why has the C-130 domain been chosen for this research when both the Defense Advanced Research Project Agency (1983) and the Air Force Studies Board of the National Research Council (1982) have identified the single-seat, combat aircraft domain as the focus of their Pilot's Associate research? This domain was chosen for several reasons. For the past six years this author has flown over 2000 hours in the C-130, and has experienced several inflight emergencies. The C-130 shares the same inflight domain with a combat fighter and also has combat mission tasking. Therefore, the domain planning knowledge and the type of goal interactions experienced by the C-130 should transfer to the fighter combat domain with minimum loss of applicability. Since the common goal of this diverse research is not purely for theoretical appreciation, the C-130 will provide a much "safer" testbed when these conceptual studies become implemented as real aircraft systems. It should be emphasized that the goal of this automation thrust is to provide an aid, hopefully intelligent, to the crewmember, not to replace him.

Operating Regulations

To understand and appreciate the C-130 emergency flight domain, the reader should first be familiar with the "normal" flight domain of a USAF C-130. This environment is closely regulated by not only the laws of physics, e.g., gravity, lift-to-weight ratios, thrust-to-drag ratios, etc; but also by the laws and regulations of the Federal Aviation Agency (FAA), the United States Air Force (USAF), and the aircraft's major command (MAJCOM). A partial list of these "man-made" regulations include Air Force Regulation (AFR) 60-16 (see appendix A), Military Airlift Command Regulation

(MACR) 55-130, Instrument Flight Rules (IFR) Supplement, Technical Order (TO) 1C-130x-1 Flight Manual, and the TO 1C-130x-1-1 Abbreviated Checklist.

The aspects governed by AFR 60-16 include, for this study, required airspeed parameters for various phases of flight, altitudes, proximity to other aircraft or ground obstacles, weather conditions and approach criteria, radio transmissions, life support equipment, and lighting requirements. "This regulation prescribes general flight rules which govern operation of Air Force aircraft flown by Air Force pilots..." (AFR 60-16, 1980). Flying outside of these regulations without permission constitute grounds for pilot violation, resulting in a severe reprimand or possible loss of flying priviledges on a temporary or permanent basis. Generally, these regulations may be violated only following a declaration of emergency, or when an appropriate authority grants such an authorization on a case-by-case basis.

MACR 55-130 further specifies restrictions on aircraft operations and outlines guidelines to follow under certain normal and emergency conditions. It contains weight restrictions, runway minimums, additional weather restrictions, mission minimum standards, etc. The pilot who violates this regulation will be reprimanded by his unit, whereas the pilot who violates an AFR 60-16 regulation is subject to violation by both his unit and the civil authorities (FAA).

The IFR Supplement functions as a source of information rather than as a directing regulation. It contains current information on aerodrome facts (elevation, runway length and direction, etc) and facilities (hangers, transient alert service, aviation fuels, emergency landing arresting gear, etc), navigation and radio aids, and other pertinent data not found in other commonly referenced publications. For example, lost communications procedures, international intercept protocals, and a millimeters-to-inches barametric conversion chart are in this reference. The IFR Supplement is included among the C-130's inflight publications on each aircraft.

The TO 1C-130x-1 Flight Manual, usually referred to as the "dash one", governs the actual "hands on" crew operation of the aircraft and it's subsystems. This manual consists of nine chapters, and has a "sister" manual called the "checklist" which is an abbreviation of the emergency procedures found in the third chapter of the dash one. Chapters one and four desribe the main and sub-systems of the aircraft, two outlines the normal procedures, while chapter five lists the operating limitations and restrictions. The final chapters discuss prohibited flight manuevers, cold weather operations and other miscellaneous information. Pilots generally commit to memory as much as possible the contents of this TO.

Other governing and flight planning regulations exist, but the intent here is to show that much of the knowledge required for inflight emergency planning has already been identified.

Pilot Training

Pilots are trained, not born. The USAF Undergraduate Pilot Training (UPT) program consists of 48 weeks of intensive education and training in the theory of flight, propulsion, navigation, weather, instrument flying, civil and military flying regulations, aerospace physiology, basic flying skills, and aircraft systems and operations. Not the least of these is the time and training effort spent on preparing the pilot candidate for the knowledge, skills and confidence to successfully "handle" emergency procedures.

The candidate is forced to commit to memory those emergency procedures deemed critical enough such that if not immediately implemented, loss of aircraft or life would most probably follow. Less critical procedures should be familiar to the pilot, but there may exist enough time for him to consult his checklist for the proper responses before further deterioration would occur. Those critical procedures committed to memory are called "bold face" because they appear in bold face type in the emergency procedures' chapters of the flight manuals. The most common "bold face" procedures dictate the actions to deal with engine fires, ejections, or unusual flight attitute recovery (spin, stall, etc).

Following the 48 weeks of UPT, the new pilots are sent to another training location to get specialized training in their assigned aircraft. Here they study and learn not only the specific aircraft operations and flight characteristics, but also the aircraft's mission or role in a particular mission. This training period can be as short as four weeks or as long as nine months, depending on the complexity of the aircraft or the mission. Again, large emphasis is placed on coping with inflight emergency procedures.

Qualified pilots face annual inflight evaluations on instrument flying procedures, reduced performance and emergency operations, and operational mission readiness. The notorious "No-notice" check-ride is a further incentive (constant threat) to "get in the books." Most MAJCOMs provide annual, emergency procedures training and evaluations in a simulator, using realistic emergency scenarios. Throughout his training and operational career, the pilot absorbs and incorporates regulations, procedures, and techniques into his thought patterns (planning model), gaining knowledge, wisdom, experience, and confidence; becoming an "expert" in the flight arena.

<u>Similarities to Everyday Activities Domain</u>

Several similarities exist between planning in the inflight emergency domain and in the everyday activities domain. Each domain requires large amounts of specific and "world" information. Getting a cup of coffee requires "knowing" about money, transportation, shopping, cooking, etc. The "Pilot Training" section of this chapter describes some of the specific knowledge required for safe operation in the flight domain.

Another important similarity between these domains is the abundant use of common sense to solve problems or to understand something. People simply "know" that airplanes can't "Pull over and stop." (Harriers and helicopters excepted) while flying. Or that if you run out of gas while flying about, you won't be stuck up there forever. People and pilots use large amounts of "native good judgement." AI researchers struggle to get machines to display this ability monopolized by human beings.

Both domains involve the rich interactions of multiple goals while performing even simple tasks. Dining out with a friend, writing a letter to your mother, changing your assigned altitude, landing on centerline, and turning on the "No Smoking" light each involve several subgoals and plans.

Differences between the Domains

The differences between the domains may be ones only of degree, but enough such that the planning model for the inflight emergency domain may be slightly less complex than the mundane activity planning model. The numerous regulations tend to constrain the propagation of flight planning options. This implies that a less complex control structure or planning model would be appropriate.

Another crucial difference is the time the planner has to construct a plan. The inflight emergency domain requires rapid planning, whereas the speed of daily activity planning probably allows some room for procrastination.

Not only must the planning be done quickly, the final plan must be right. Outcomes of the planner in the inflight emergency domain are generally more crucial than those from an everyday activity planner. In other words, there is less room for error or marginal plans when the loss of lives and property are the likely consequence of a "bad" plan.

Taxonomy of Goals

Because so many diverse goals exist in the inflight emergency domain, a taxonomy or classification of these goals helps to define an appropriate planning structure to deal with them. A list of these goal classifications with a short description follows.

At the pinnacle of the goal taxonomy sits the Human Resource Preservation Goal (HRPG). This goal type indicates that the highest priority of the planner is that of preserving human life and or limb. During the resolution of a conflict with a lesser goal, this goal will have priority. The Flight Manual posts WARNINGs throughout its chapters identifying actions which are dangerous to the crew and passengers. Smoke or noxious fumes in the cabin or too little oxygen are examples of situations which are direct threats to the crew and passengers.

The second goal type is the Material Resource Preservation Goal (MRPG) which uses plans which try to prevent the loss or destruction of material items like tires, engines, wings, cargo, the aircraft, or the PIPA itself. Obviously a hierarchy inside a type becomes necessary since the planner may have to decide between the sacrifice of some engine-fire extinguishing agent and the engine itself. Or in the HRPG area, it must know that the value of 25 people's lives is higher than the value of five people's lives.

The third goal classification is the Mission Accomplishment Goal (MAG). That is, get the job done as long as it is not at the expense of HRPGs and MRPGs. There are exceptions to this and the planner has to be aware of these. For example, some missions are so important that even high risks to personnel and material are tolerated, even following degradation to aircraft system performance. Every pilot has heard or experienced a "There I was..." war story.

Included in this goal type are the Flight Phase Goals (FPG). A flying mission divides into several phases such as preflight, taxi, take-off, climb, cruise, descent, landing, and an assortment of specific missions (airdrop, air-refueling, bombing, strafing, photo-recon, etc.). Each of these phases may have its unique goals associated with that portion of the mission. For example, only during the air-refueling portion of a mission does the Flight Engineer closely monitor the "Hose in refueling range" caution and warning lights.

Another more subtle goal category involves the professionalism of the pilot and his desire to perform well while following the rules, regulations, and informally established modus operandi. The debate of whether or not this desire is motivated by wanting to avoid violation or punishment as opposed to wanting to do well for higher motives will not be held here. This goal group has been called the Maintain Status Goal (MSG).

This next goal classification is the least quantifiable and most difficult for incorporating into a planner. It is the Ulterior Motive Goal (UMG). It is the "real" reason the pilot may want to do something masquerading as another, more acceptable goal. Consider the example where following the inflight shutdown of an engine, the pilot chose to proceed to the closest airfield, rather than turning back several miles to an airfield offerring much better maintenance service. His justification stressed expedience and safety, "What if something else had gone disasterously wrong on the way to the further airfield?" Actually; however, the base commander at the base he landed was an old buddy of his he

HUMAN RESOURCE PRESERVATION GOALS

MATERIAL RESOURCE PRESERVATION GOALS

MISSION ACCOMPLISHMENT GOALS
FLIGHT PHASE GOALS

MAINTAIN STATUS GOALS

ULTERIOR MOTIVE GOALS

Figure 2. Taxonomy of Flight Domain Goals

wanted to see... These secret goals tend to confuse the planner when it tries to figure out the pilot's actions. The next time the planner is in a similar situation, the pilot may elect the further base and really puzzle the PIPA.

This goal taxonomy (Figure 2) is not listed in order of importance because there can not be an explicit ordering of these goals. Their priority would be circumstantially decided at the time of planning and acting.

Goal Conflicts

The inflight emergency domain is rich in multiple goal conflicts (see Figure 3) because of the abundance of active goals at any time. This list of goal conflicts indicates some of the difficulties the planner and the planner designer face. The list is not in significant order.

Crew and aircraft safety frequently conflicts with mission objectives. For example, to deliver their cargo of ammunition and medical supplies, the crew has to penetrate several miles into enemy held and well defended territory. The conflict lies in the fact that a precondition for keeping the crew and aircraft safe is "avoid areas of known hostilities or danger," but mission orders specifially require flight into just such areas to accomplish the airlift mission.

Frequently the crew will relax the safety goal for personal reasons to support the mission. Perhaps their friends are the ones who need the supplies, perhaps there is

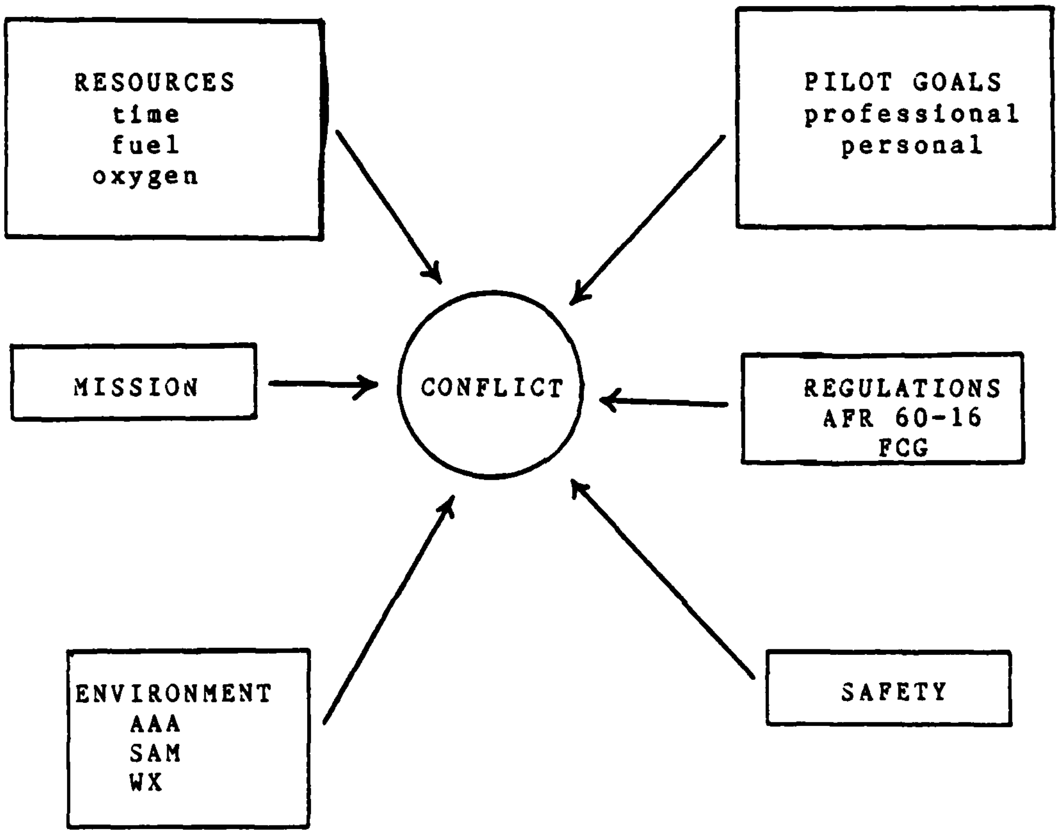


Figure 3. Goal Conflicts in the Flight Domain

some special reward associated with the successful outcome of this particular assignment. Maybe they have no choice!

Conflicts of the form "regulations versus personal goals" arise when for instance, a pilot wants to fly lower than regulation minimums, or faster than allowed maximums. Personal goals will frequently conflict with safety regulations (percieved to be too conservative) and occaisionally with mission goals.

There are goal conflicts imposed by resources such as fuel, oxygen, time, oil, thrust, runway lenght, maintenance, or spare parts. An airplane cannot fly for six hours with only three hours of fuel on board. The goal to provide seven people with continuous oxygen cannot be met with just five oxygen masks. The goal of having sufficient oxygen in an unpressurized aircraft and the goal of flying safely above the tops of the mountains result in a conflict.

Despite this variety of goal types and their conflicts, pilots cope quite well for the most part. However, if a machine pilot aid could prevent even a small percentage of the future accidents, the return should far exceed the investment. The following two examples help explain how pilots plan during an emergency and lend insight into how a machine planning structure may be designed.

A Rapid Decompression Emergency

Assume a C-130 with a crew of five is on a mission to fly 25 passengers from Denver, Colorado to Spokane, Washington. While at 25,000 feet over the Rocky Mountains with tops up to 14,000 feet, the aircraft experiences a sudden loss of cabin pressurization, aptly called a rapid decompression.

Immediately the pilot commands his crew to don their oxygen masks and check in with him on intercom. He initiates a descent to the Emergency Safe Altitude and tells the copilot to declare an emergency with the enroute controller. He has the flight engineer figure out what caused the depressurization and asks the loadmaster how the passengers are doing since they have no supplementary oxygen and were probably quite alarmed or perhaps injured in the decompression. The pilot and the navigator chose a route allowing the lowest altitude and a destination with adequate medical facilities within the shortest flight time. The crew pursues efforts to make the passengers as comfortable as possible until landing.

How did the pilot arrive at the plan he used? A look in the inflight checklist (Appendix A) under Rapid Decompression shows only "OXYGEN -- As Required" and "Descent -- As Required." Certainly not enough information if a machine is to provide intelligent help to a pilot. To know if OXYGEN is required, the pilot has to know the appropriate contents of AFR 60-16 (Appendix A), which in effect says anytime the cabin altitude rises above 10,000 feet, the crew will don their oxygen masks with the regulators set to "ON" and "100 PERCENT."

Concerning the "Descent -- As Required," AFR 60-16 also states for passengers without supplemental oxygen (normally the case in a C-130), the aircraft will be flown to below 10,000 feet MSL. However, the pilot's common sense told him if he were to fly this low, they'd crash into the mountain. AFR 60-16 additionally states if an altitude below 10,000 feet is not possible, that 13,000 and below for up to three hours is allowed. Unfortunately, mountain tops above this altitude also precluded this option.

But the pilot knows that even though he cannot comply with the regulations, the lower he can safely fly, the more oxygen available for his passengers. He also knows that the Emergency Safe Altitude (ESA) provides 2000 feet of clearance over the mountain tops within a 22 mile wide flight path. Therefore, though perhaps unable to see the terrain because of weather, he knows he can provide the lowest altitude with safe terrain clearence.

So what actual planning strategy did the pilot use? First of all, as soon as he noticed the depressurization, he recognized the goal of getting oxygen for himself, his crew and passengers. He employed the normal emergency procedure or "canned" plan of donning oxygen masks for his crew. His earlier preflight confirmed the preconditions "have masks" and "oxygen system full" were met. In trying to apply the "canned" plan for the passengers, he saw this goal conflicting with the goals to preserve Human and Material Resources by not flying/crashing into the mountain. The alternative "canned" plan produced the same conflict. He recalled a plan which allowed the passengers to use the emergency smoke masks with an oxygen bottle with the precondition that each passenger have one. But only five smoke masks are on board, so this plan gets rejected for lack of a fulfilled precondition. The plan to make this precondition true would require going back to a base to pick up more masks. The smoke mask plan is rejected.

The pilot reasons that the goal "having oxygen" cannot be satisfied as required by regulation, but he knows that people need oxygen to live. He also knows that the lower he flies, the more oxygen available. So he flies as low as safely possible until they can reach a suitable destination.

The planning process involved detecting a goal, finding a plan to reach the goal, incorporating large amounts of regulations and world knowledge, testing the plans, abandoning unreachable goals for attainable ones, and repeating the cycle as necessary.

A Wing Fire Emergency

The second emergency example involves a crew on a mission to fly cargo from San Antonio, Texas to Little Rock, Arkansas. At 22,000 feet, the crew of five are alerted by a fire warning light for the number three engine. The pilot directs an emergency engine shutdown in accordance with the checklist. The pilot detects the goal, recalls the appropriate plan, mentally confirms it, and executes it.

The shutdown is successful, but a wing fire breaks out. The pilot immediately begins the "Wing Fire" checklist procedures of lowering the nose and adding power to rapidly increase airspeed, simultaneously slideslipping the aircraft to prevent the fire spreading toward the fuselage. At 4000 feet above the ground and at 350 knots, the fire finally extinguishes. The pilot begins a climb to trade airspeed for altitude, but when slowed to under 250 knots, the fire reignites. The aircraft is already too low to be able to gain much airspeed by descending, and on three engines cannot accelerate rapidly. The fire continues to spread along the wing towards the fuselage.

With no parachutes on board, the preferred plan "BAILOUT" is rejected. The pilot sets up for an emergency forced landing, with or without a runway beneath them.

The planning strategy in this example was again to detect a goal (put out wing fire) based on an observation (wing on fire), find the plan for achieving that goal, (wing fire checklist plan), and test the plan and employ it if it would appear to work. It did in the first wing fire instance, but not the second. The plan to increase airspeed to snuff out the fire could not be carried out on three engines at a low altitude.

The pilot now realizes that he can not attain the goal to put out the wing fire, and now recognizes the danger he and his crew are in, and now detects the goal of perserving self and crew by abandoning the burning aircraft. This can be accomplished by either bailing out if parachutes are available, or by an immediate forced landing and evacuation of the plane. In either case, the pilot has to incorporate a large amount of domain knowledge and multiple goals into his planning strategy.

Other Emergencies

The over fifty emergency procedures described in chapter three of the Flight Manual (TO 1C-130x-1) represent some of the most common or frequently occurring situations. With systems as complex as modern military aircraft, literally thousands of causes of malfunction require some procedures to minimize the damage and possible loss of life. The Appendix C contains the transcripts of three other emergencies in addition to the ones discussed here. The point of their inclusion is severalfold.

These examples show that given a certain situation, the pilots recognized the same goals, and produced similar plans to reach the goals. They also show that slight variations in plans can still achieve the goals, indicating that there are more than one plan, and perhaps no single optimum plan.

These examples also show that while the Flight Manual contains the recommended procedures for specific emergencies, it does not provide quidance for all combinations of possible emergencies. Hence the statement at the beginning of the manual about it not being a substitute for sound judgement on the part of the pilot and crew.

These examples also show the protocals used by pilots in the planning process, protocals which fit well with the planning approach proposed by Wilensky.

IV. Conceptual Design of the Planner

Introduction

This chapter discusses the requirements of a Planner in an Intelligent Pilot Aid (PIPA), discusses the components of the planning structure, and presents a paper example of how the planner would operate.

Planner Requirements

Autonomous Goal Detection. An acceptable PIPA in the inflight emergency domain requires several capabilities. The planner must have autonomous goal detection ability, that is, it must not rely on inputs from the pilot to know when and what goal to begin planning for. The nature of many inflight emergencies does not allow time for the pilot to explain to the PIPA that because the "fire warning light on number three" is illuminated, it now has the goal to "put out the fire on number three." It is critical for the PIPA to infer its own goals and begin its planning process as quickly as possible.

Explanation and Understanding. The PIPA is required to understand the pilot's questions and commands and has to be able to explain its answers if so asked. It should entertain goal and planning questions posed by the pilot or other crew members. Conducting "What if..." sessions with the PIPA could provide valuable training to newer crewmen as well as provide a good refresher for the "old heads." The PIPA uses large amounts of domain knowledge for planning and should share it when requested by the user. The following dialog demonstrate the types of questions and answers the PIPA should support:

PILOT: What is your goal if you see the engine number three fire-warning-light illuminate during the cruise portion of flight?

PIPA : Put out engine number three fire.

PILOT: How?

PIPA : Condition lever, number three---feather.
Fire handle, number three---pulled.
Agent, number three---discharged.
Cleanup.

PILOT: Why?

PIPA : An engine fire is a mandatory engine shut-down situation and those steps are the directed procedures.

PILOT: I mean, why do we want the fire out?

PIPA : An engine fire allowed to burn can cause extensive engine damage and potential total destruction of the aircraft. I can be more specific.

Note here that the PIPA seems to have two ways to understand the question WHY? The cause for the first response is found verbatim on page 3-3 in the Flight Manual, "If any of the following conditions occur, shut down the affected engine...Engine fire..." It is similar to a mother telling her son to do something "because I said so!" However, when the PIPA recognized that the PILOT wasn't satisfied with its first answer, it then gave a brief projection of what would happen if he did not shut down an engine on fire. If requested, the PIPA could provide much more specific information on the effects of an engine fire left to burn.

AI researchers in the area of natural language understanding have devoted much effort towards solving some of the problems of representing "meaning" in a computer. For example, you're seated at the counter of a truck-stop cafe having lunch, when the man nearest you leans your way and says "Hey, buddy, can you reach the salt?" You respond by picking up the salt shaker and handing it to him, understanding what the man actually meant. Had you looked at the salt shaker, made a mental measurement, and then turned to the man and answered honestly, "Yes," you might require some first-aid, depending on his temperment and size.

Continuing the dialog...

PILOT: What regulation governs engine shut down?

PIPA : Your Flight Manual, Chapter Three, pages 3-6.

PILOT: Are there any warnings associated with this procedure?

PIPA : Yes.

PILOT: Go on.

PIPA : Engine number two now provides the only engine air inlet anti-ice detection and control. In the event of its failure, there is no automatic engine anti-icing. Danger is ice build-up and subsequent ingestion and engine damage.

PILOT: Can a restart be attempted?

PIPA : It is not recommended to restart an engine shut down for fire, unless an emergency of higher priority exists.

Chapter five discusses the implementation of the PIPA and includes a section on how this explanation capablity is incorporated.

Conflicting Goals. The planner must handle conflicting goals, combining and manipulating plans efficiently to accomodate as many goals as possible. For example, it may have to maintain airspeed, altitude, crew comfort, and heading, while conserving fuel and observing applicable flight regulations. One difficulty of multiple goal manipulation is that the plan steps for one goal may undo a precondition for another goal. Sussman noticed this and called it the problem of "prerequisite-clobbers-brother-goal," during work on his model of skill acquisition, HACKER (Sussman, 1975). Generally speaking, a goal conflict exists when the realization of one goal interferes with the realization of another goal.

For instance, suppose the precondition for the goal "fly to destination A 1000 miles away" is "have enough fuel on board" which equates to 2000 pounds based on an airspeed of 200 miles per hour." Suppose now the pilot has the goal "fly airspeed of 500 miles per hour" which uses a fuel-flow rate of 1500 pounds per hour. If this plan is used, the aircraft will flame out in an hour and twenty minutes, never reaching destination A. The second goal can be met, but at the sacrifice of the first one. The plan of the second goal negates a precondition of the first goal. The planner must now do what it can to effect the most desireable solution, either replan or change the circumstances.

What options should the planner have when confronted with multiple goals? What does the domain demand? The inflight emergency domain requires the planner to be able to recognize goal priority (see Chapter three) and plan accordingly. For example, the pilot tells his PIPA that he wants to conserve fuel and avoid enemy radar detection. Normally, the higher the altitude the more efficient the fuel burn rate, and the lower the altitude the harder to detect by radar (terrain masking) are the simple rules of thumb. The planner should recognize that it is more important to remain undetected and construct the plan "fly low level using terrain masking at optimum airspeed of xxx knots." The cost of getting shot down is higher than the cost of the extra fuel it burns at the lower altitude.

Occasionally multiple goals can be satisfied with one plan. For example, a C-130 departing Seoul, Korea is cleared to FL190 (19,000 feet MSL) on its departure for Tokyo, Japan during an airlift exercise in December. At

FL190 the C-130 is slightly below the tops of the clouds, picking up rime icing on its wings, tail and radome. The pilot recognizes three goals: 1) get out of the icing conditions, 2) conserve fuel (they are below optimum cruise altitude), and 3) take advantage of the winter jet stream which traditionally flows easterly in the winter months and desends to the FL200 region. He requests and recieves from the Seoul route controller a climb to FL230, thereby achieving three goals. Wilensky (1983a:53) calls this goal overlap when a plan supports multiple goals.

Essentially the goal interactions can be grouped into three general categories. The first catgory involves no interaction, that is each plan can be executed independent of and have no effect upon the other. For example, the number one generator-out light illuminates the same time the loadmaster calls out that a cargo compartment window outer pane just cracked. Other than the generator goal receiving more attention, these particular goals involve no interaction the planner has to consider.

The second general type of interaction involves goals whose plans overlap as discussed above. An example in the emergency domain demonstrates it further. During a high speed low level training route, the C-130 encountered a flock of migrating waterfowl and sustained multiple bird-strikes. Several goals are activated. Get above the other birds possibly in the area, begin paying more attention inside the aircraft (low-level flying requires lots of attention outside the aircraft), reduce speed to lessen structural loads on the aircraft, make a bird-strike radio call to warn others flying in the same area (higher altitude will permit wider broadcast), and set up for return to base are goals which use the same plan step "begin an immediate climb."

The third type of goal interaction involves goal conflicts and has two subtypes. The first subtype allows the two (or more) conflicting goals to be met by finding or building a single new plan to meet the goals. For example, during the cruise portion of a flight, a crew may have the goal fly as high as possible for fuel efficiency and the goal breathe adequate amounts of oxygen. If cabin pressurization fails, the plan to descend to lower altitude would interfere with the fuel efficiency goal. So the plan to breathe supplemental oxygen is a single plan which can allow both goals to be maintained.

The second subgoal type involves incompatable goals where one of them must be abandoned. For example, a C-130 flying at 1000 feet AGL (above ground level) loses two engines (flameout). The aircraft weighs 130,000 pounds so the pilot begins dumping fuel immediately since 120,000 pounds is the maximum weight limit for safe two-engine operation. The goal to "follow regulations" which state "fuel dumping prohibited below 5000 feet AGL" had to be abandonned in order to maintain safe flight. In other words, the Human Resources Preservation Goal with a plan calling for the immediate weight reduction in the form of dumping fuel, took priority over the Maintain Status Goal of following regulations, which had to be abandonned in this instance.

Dynamic Environment. The planner needs to accomodate a dynamic environment of changing missions, weather, terrain, airspace restrictions, engine performance, or flight envelopes. For example, during the cruise portion of a flight the PIPA notices that the aircraft gross weight now permits a climb to the next higher cruise altitiude, the plan which supports the goal of "use fuel resources wisely." Then, just as the PIPA was going to inform the pilot of the recommended altitude change, it notices that the oil pressure on number one is below operating limits. The PIPA must now because of the changed circumstances put the climb goal on hold or abandon it until it can take care of the more important goal "prevent further damage" concerning engine number one.

The direct effect of the dynamic environment is realized in the changing priority of the goals, and hence in the plans chosen to accomplish the goals. Basically the environment dictates the goal priorities.

Simulation. The planner has to test its proposed plan before presenting it to the pilot or performing the actions specified in it. Testing should involve some method of simulation or projection of the plan into the future, watching to see if the plan will achieve its goal without conflicting with other active goals. If the plan does not work, the planner should learn from this projection how it might improve the plan or suggest a new one. If the new plan would remove another plan's precondition or lacks one of its own, the planner should first try the least expensive way to fix the problem. All plans must be tested, even recently used ones, just because a plan worked an hour ago, doesn't mean it will work now. As an example, consider the wing fire emergency example discussed in chapter three.

After the plan checks out satisfactorily and the pilot initiates it, the PIPA's job continues. The PIPA must constantly monitor the consequences of the actions carried out. It may have to alter the plan during execution because of new conflicts or unpredicted world events. In this sense, the planning process is not actually complete until the end of the planned steps.

PIPA Components

The planner for the flight emergency domain consists of four major components (Figure 4) or software modules. The divisions between the modules may be somewhat indistinguishable at times because of the closeness of their interactions. The following paragraphs describe these components.

<u>The Goal Detector</u>. The responsibility to monitor or watch for something to happen belongs to the Goal Detector. This module maintains lists of items or states it has an interest in. Each phase of flight (Chapter three) has goals particular to that portion of the mission. For example, ordinarily the planner will not be looking for SAMs (Surface to Air Misslies) during the "taxi" portion. The same observation may even require different treatment if occurring during different flight phases. An engine fire warning light requires a different plan on "takeoff" than it does during "cruise." Associated with each observation is a goal, such that, when the observation is observed true, the goal is automatically known. For example, during the cruise phase the goal associated with the number three engine fire warning light "ON" is the goal "put out engine fire." In effect then, the Goal Detector is a list of goals with the observations which trigger them. This portion of the planner contains the simplest structure of the PIPA.

In addition to monitoring the "real-time" environment, the Goal Detector also monitors the Plan Projector's "simulated" environment, helping spot goal conflicts and other weaknesses of a proposed plan. Goals detected during this simulation are treated just like the goals encountered in it's real domain.

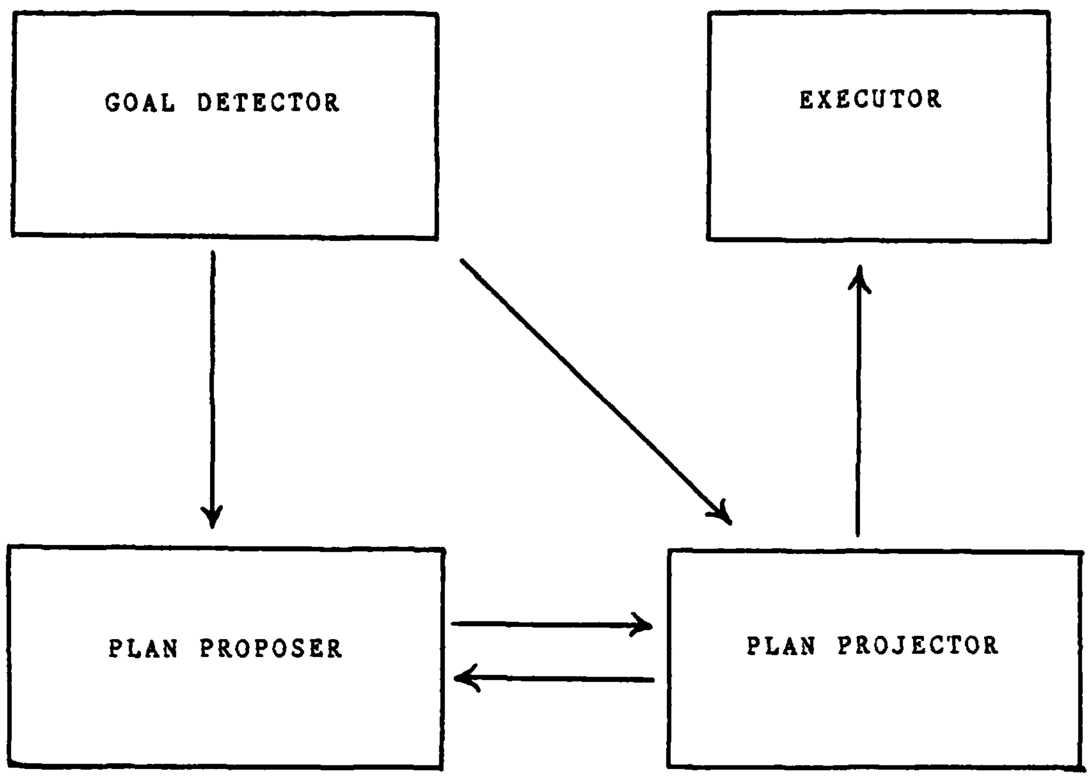


Figure 4. Components of the Inflight Domain Planner

The Plan Proposer. The Plan Proposer, more complex than the Goal Detector, recieves a goal from the Detector and looks to see if it maintains a plan for it. The goal, for example, to "put out the engine fire" would have the plan "Emergency Engine Shutdown" associated with it. Some goals may have more than one plan associated with it, but the first plan will generally get proposed first. Information about the plans is available to the Plan Proposer so some intelligent plan chosing can occur. For instance, suppose a crew member has the goal "have oxygen" and two of the three available plans require the use of an oxygen mask. After submitting the first plan to the Plan Projector, the Proposer is told this plan fails because an oxygen mask is unavailable. The Proposer now submits the plan not requiring use of an oxygen mask. If none of the known plans are successful, the Proposer must reference its model of the oxygen system and try to determine if any other plan can be used to get oxygen.

The Projector. The Projector accepts the proposed plan and projects it into the future beginning with the current world state. The effects and defects of the plan can be studied, offering a chance to improve the plan if necessary. During this projection other goals which may pop up, detected by the Goal Detecter, are turned into plans by the Proposer and factored into the simulation. The Projector must have knowledge of the aircraft, its support subsystems, and the aircraft's real-time environment. It has to know if a failure in the simulation is caused by the plan or some other faulty process already in progress. These requirements are recognized to be quite difficult, but regardless of these difficulties, this capability remains a necessity for a flight planner if it is to be truly helpful in the aircraft domain. Therefore, the Projector must test the plans by simulation, passing the good ones to the Executor and replanning the bad ones.

The Executor. The Executor is the fourth component of the PIPA. It functions as the communicator to the pilot and as the assistant controller of those flight subsystems entrusted to its care. After the Projector passes the approved plan to the Executor, the Executor passes this plan to the pilot or other designated crew members. In actuality, the Executor is an interface to the crewmembers.

Meta-planner

The meta-planner is the theme implicit in the structure which guides the planning process in the PIPA. Not all planning problems can be solved using the same planning steps; therefore, the nature of the plan interactions should dictate the next planning step. Conflicts should initiate some actions to solve it, the lack of a plan should cause one to be constructed, etc.

The sequence of events guided by the meta-planning structure is as follows: the Detector notices something of interest and passes the goal to the Proposer. The Proposer finds the plan for the goal and passes it to the Projector. The Projector projects the plan into the future and the Detecter possibly sees a conflict, raising the goal to resolve conflict. The Proposer then looks to see if it has another plan which wouldn't cause the same kind of conflict. If it has one, that plan is tried, if not, the Proposer tries to find a plan by reasoning about its model of the domain. Once found, the plan is tested by the Projector. If successful, the planning is nearly finished, else the Proposer looks to see if the circumstances themselves can be suitably modified and replanned. If not successful, the goal gets abandonned.

Emergency Example

The PIPA will use the Rapid Decompression example from the previous chapter for a paper example of how it could work the goal conflict problems.

During its cruise at FL250, the Goal Detecter notices the pressure altitude climb rapidly from 7000 feet to 25,000 feet. As the pressure altitude passed 10,000 feet, it sent to the Plan Proposer the goal "get oxygen to crew and passengers." The Proposer sends the plan "crew don oxygen masks" to the Plan Projector which in turn tests the plan for acceptance. The projection shows that the preconditions "crewmembers have masks" and "sufficient oxygen available in system" (for planned duration based on original flight plan) are satisfied (these items were verified by the crew during mission preflight also), so the Projector passes this plan step to the Executor. The Excutor, in turn, tells the pilot the tested plan, ready to answer any questions he may ask.

The Proposer passes the plan "descend below 10,000 feet" as its plan for the passengers to get oxygen. The Projector projects the plan into the future in the hypothetical model of the current state of affairs. The Detecter sees the aircraft flying toward the ground (mountains) and detects another Human Resources Preservation Goal called "maintain safe altitude above the terrain." The Projector rejects this goal and returns the plan to the Proposer with the message that flying below 10,000 feet is not possible because of terrain. The Proposer checks its other available plans, ommitting any requiring flight below 10,000 feet. It finds "descend below 13,000 feet for a period not to exceed three hours above 10,000 feet."

The Projector also projects this plan and it too fails for the same reason the first one did. The plan and message come back to the Proposer, and it sends its last plan "use smoke masks and oxygen bottles." The Projector tries this plan and discovers the precondition "each passenger have own mask and bottle" fails because there are 25 passengers and only five smoke masks. The goal "get 20 smoke masks" pops up, but the plan "call Life Support (smoke mask managers)" failed. The Plan Proposer has tried each of its known plans, so now the Proposer must try to devise a novel plan by reasoning about the relationships of oxygen in the flight domain.

It finds the relationship between oxygen and altitude and concludes that the aircraft should be flown as low as possible, taking into account safety of flight items such as mountain tops, severe weather, and other traffic. The Proposer then passes the new plan "fly as low as safely possible" to the Projector. The plan is simulated and found to be okay, with the Navigation package providing the correct ESA altitude for the Projector.

During the portion of the flight while the aircraft is flying at the ESA, the Detecter is instructed to watch for when the terrain will permit a descent below 10,000 feet and subsequently fulfilling its original goal to descend below 10,000 feet MSL.

The next chapter discusses how this planner can be partially implemented using current AI programming techniques. Though the purpose of this research was not to implement a complete planner, enough portions of it were built to determine the appropriateness and implementability of Wilensky's planning theory in this domain.

V. Implementation of the Planner

This chapter discusses how the conceptual design discussed in the previous chapter can be implemented using contemporary AI programming techniques. The following paragraphs describe what technique was used to construct each component and why that particular implementaion was chosen. The Rapid Decompression example is implemented.

Programming Language

The author chose Franz LISP, Opus 38 dialect, installed on a Dec VAX 11/780 as the implementation language for this planner model. LISP was chosen because 1) several useful LISP subroutines and functions already existed in the Air Force Institute of Technology Artificial Intelligence Laboratory program library, 2) this program will be transferred to a Symbolics LM/2 LISP machine, 3) and also because LISP supports PEARL (Package for Efficient Access to Representations in LISP), a representation AI programming language (Wilensky, 1983b; Deering, 1982). Implementing this planner in PEARL should significantly improve efficiency as its size increases.

Details on programming in LISP, the preferred AI language in the United States, can be found in Winston (1981, 1977) and Wilensky (1984). Charniak (1980) provides advanced LISP programming techniques.

The Goal Detector

The Goal Detector is a rule-based, forward-chaining, data-driven production system consisting of a control structure, a data base, and a knowledge base. The control structure of this system is quite similar to the one used in Winston's (1981:240-249) animal-identification world. The knowledge base contains the formatted production rules which are divided into two parts. The IF, or antecedent, part which contains the item or state the Goal Detector is tasked to watch for, and the THEN, or consequence, part which consists of the goal associated with the detected IF part. A simple example looks like this:

```
RULE17
    IF-I-DETECT (FIRE-WARNING-LIGHT-ON-ENGINE-ONE)
    THEN-MY-GOAL-IS (PUT-OUT-FIRE-IN-ENGINE-ONE).
```

Each of these rules represents knowledge contained in the technical orders and flying regulations. Some of the observation/goal relationships are based on human common sense, rather than explicit declarations in the flight manuals. For example, the "dash-one" doesn't have a procedure describing what to do in case all four engines flame out. A pilot's common sense will tell him to attempt to restart at least a couple of the engines. Since a computer has no common sense, all rules pilots associate with common sense have to be explicitly declared, and where necessary, the physical laws which apply (gravity, fuel consumption, etc.) have to be incorporated. The basic appearance for a production rule in the inflight emergency domain is:

```
RULEXX
    IF-I-DETECT (SOMETHING-JEOPRODIZING-SAFETY)
    THEN-MY-GOAL-IS (TO-PROTECT-MYSELF).
```

The data base contains the information about the currrent state of the aircraft, its phase of flight, and its environment. The different phases of flight, "landing" versus "cruise" for example, may involve goals specific only to that portion of the flight, so the Goal Detector monitors only those necessary at the time. This information is stored as dynamic lists, growing, shrinking, and changing as the environment dictates.

The control structure is a forward-chaining procedure which searches through the production rules and data bases, until an IF part (observation) is found true. The THEN part (goal) gets added to a currently-under-consideration goal list and in turn gets passed to the Plan Proposer. The search process continues to the end of the production rules list and repeats. Essentially, this portion of the planner acts as the monitor, watching for those items which are a threat to flight safety.

After an observation is discovered true, its rule is removed from the active search list until it becomes applicable again, perhaps a few minutes later or perhaps not until the next flight after proper maintenance can be performed. For example, the observation "Right-wing auxilary-fuel-tank quantity guage --- Off-scale high" caused by an electrical discontinuity requires the flight engineer to remove electrical power to the fuel probes and guage for that tank by pulling a specific circuit breaker. The guage reading will remain in the "off-scale high" position. If this observation were left in the monitored list, this observation would be repeatedly found true even though it is now unnecessary.

Malfunctions which are observed and corrected, such as "propellar RPM out-of-limits," would be added back to the list so the Goal Detector can be watching for it to happen again. The function

```
(move-rule rule  *list1* *list2*)
```

which removes a rule from list1 and places it into list2, is used to place the appropriate rules into the proper lists.

The Plan Proposer

The Plan Proposer, responsible for maintaining plans and planning information, is implemented in a frames system. Minsky (1975) says a "frame is a data-structure for representing a stereotyped situation" like being in an aircraft cockpit or flying a routine mission. Other information can be included in a frame such as when to use or how to use the frame. Frames have also been called schemas or scripts and essentially are ways to partition knowledge into machine manipulatable structures.

A frame can be thought of as a network of nodes and relations with the top levels consisting of fixed representations. The lower levels consist of slots or specialized instances of the attributes. Consider the following frame as a parent or prototype frame for an aircraft:

AIRCRAFT Frame with

Composed-of	=	Fuselage, wings, engine(s), tail
Requires	=	Aviation fuel, oxygen
Capable-of	=	Powered flight
Operated-by	=	A pilot (at least)

and this frame as a child or an instance of the AIRCRAFT frame:

F-16 Frame with

Instance-of	=	AIRCRAFT
Number-of-engines	=	1
Type-of-propulsion	=	turbo-jet
Number-of-crew	=	1
Type-of-mission	=	interdiction.

These frames begin to reveal the powerful, frame system feature of property inheritance. The F-16 frame inherits the properties of the parent frame, AIRCRAFT, through its "Instance-of" slot. If the question "Does an F-16 have wings?" arises, the default answer is "yes," because in the

AIRCRAFT frame resides the knowledge that an aircraft has wings and since an F-16 is an aircraft, it too has wings.

Several realizations of Minsky's theory of frames exist today. Some are more specialized such as scripts by Schank and Abelson (1977) and frames by Charniak (1977); while others are general knowledge structures such as Knowledge Representation Language (KRL) by Bobrow and Winograd (1977) and Frame Representation Language (FRL) by Roberts and Goldstein (1977). The frame representation used for this IPA planner is influenced by FRL, XRL (Unknown Representation Language) by Charniak, Riesbeck, and McDermott (1980:178-192), and Cross (1983:42-45).

The Plan Proposer consists of a network of plan frames containing the goal, the plan(s) known to satisfy the goal, plan preconditions, and other useful planning information. An example of a plan frame looks like this:

```
(frame crew-have-oxygen-plan   isa plan with

      (goal            = crew-have-oxygen)
      (checklist-plan  = crew-don-oxygen-masks)
      (preconditions   = crew-have-oxygen-masks
                         oxygen-in-system)
      (warning         = extinguish-smoking-material)
      (reference       = dash-one-chapter-three)).
```

Additional slots can be created and added to the frame for other purposes such as keeping a history of the successes or failures (and why) of the plan or the last time it was used. Each of the above slots have a slot name, an aspcect, and a value. The "=" symbol is the aspect and means that, for example, the slot name "warning" has the value "extinguish-smoking-materials." A slot name's value could also be another frame, as in the AIRCRAFT and F-16 frames example.

Occasionally the value of a slot name cannot be explicitly declared in the property list with the "=" aspect. Another powerful frames feature overcomes this problem by attaching programs to the data structure, called "procedural attachment." For example, the F-16 frame could have the slot

(total-aircraft-weight if-needed get-total-weight)

which means if the total aircraft weight is needed then the program get-total-weight calculates and returns the sum of the weights of the basic aircraft, the fuel, the bombs, rockets or missiles, and any other added items. The aspect is the "if-needed" part, and the "get-total-weight" is the program which retrieves the sum. The "if-needed" aspect is a member of the general class of procedural aspects called servants because it waits to be asked (told) to do something.

A second general category of procedural attachment programs, called "demons," are activated automatically whenever information is added to an instance. An "if-added" demon is used to specify satifactory values as new slots are defined. Other aspects and their descriptions can be found in Cross (1983:43-45).

The Plan Projector

Also a frames system, the Plan Projector implemention performs a quasi-simulation of the plan by comparing key aspects of the plan to known restraints or limits. For example, in the plan using the step "descend to 10,000 feet," only aircraft altitude and terrain elevation were considered. As long as the aircraft altitude remained well above the terrain elevation, there was no conflict. However, if the terrain elevation were higher than 10,000 feet the aircraft would crash, hence a conflict between the goal calling for this plan and the goal "fly safe." The effects of the lower altitude and denser atmosphere on true airspeed, angle of attack, and fuel effeciency were not examined unless specifically involved with another goal.

The procedural attachment demon, if-added, is used as the aspect in the following slot to test the plan for satisfied preconditions and conflicts with other goals before it is added to the active-plans list:

```
(plan  if-added (and (preconditions-met)
                     (no-goal-conflicts))).
```

The functions "preconditions-met" and "no-goal-conflicts" compare items in property lists and signal the acceptance or rejection (true or false) of the proposition. If the plan is accepted, its newly instantiated frame is passed (in effect) to the Executor. Actually, the program justs prints out the verified plan to the terminal screen. If the plan is rejected, the next plan (if it exits) is taken from the plan list and tested in the same way. If no known plan achieves the goal either due to lack of precondition or goal conflict, the goal is abandonned. This implementation was unable to offer new plans or to reason about how or why the goals came about.

The rapid decompression example at the end of this chapter will show this implementation, but first a discussion of some approaches that may be used in plan simulation, projection, and evaluation. The Projector's responsibility is to test plans for successful goal attainment. It must represent the "current emergency state," apply the plan to it, and watch for any weaknesses in the plan. Essentially then, the plan can be understood as "successful" or "unsuccessful" only in light of the whole situation. The simulation becomes merely an understanding of the effect of the plan. For the inflight emergency domain, what representation(s) will provide this understanding?

Commonsense Algorithm

An approach to represent the basic data structure for modeling human cognition was proposed by Rieger (1975) in his concept of the Commonsense Algorithm (CSA). This structure "is defined by specifying a set of proposed cognitive primitive links which, when used to build up large structures of actions, states, statechanges and tendencies, provide an adequate formalism for expressing human plans and activities, as well as general mechanisms..." (Rieger, 1975:1). The CSA can express both static and dynamic subjects. A static subject could be an aircraft itself just parked on the ramp, while a dynamic object could be a rotating, variable-pitch propellor on a C-130 aircraft.

If each system of the aircraft is "described" by a CSA representation and then linked together into a network-like structure called AIRCRAFT, it could simulate aircraft system functions showing the causal relationships between the components.

Rieger describes his CSA in detail and shows an example application using the "operation of a reverse-trap toilet." This example highlights the importance of feedback in a system. Many aircraft systems are controlled via some feedback mechanism, or are providing information to the pilot using such a mechanism. By representing a system in a CSA, applying a plan to it, and observing the effects, the essential parts of the simulation are accomplished.

Underwood (1983:302-305) has modelled several physical mechanisms of a nuclear power reactor in a CSA network model. He uses the model as a consultant for diagnosing nuclear power plant problems. Both normal and abnormal states and operations are represented, along with diagnostic rules for troubleshooting. The model can be given a problem description and asked to diagnose the cause. Or it can be asked why the automatic control system would perform an action it took during an abnormal event.

Underwood's model is not wholly CSAs, but also incorporates the use of a forward chaining control strategy, integrating the diagnostic rules into the CSA net. The success of the system is to be experimentally validated by using the methodology used to test MYCIN (Underwood, 1983:305).

A PIPA could use a CSA network of certain systems to help it propose new plans after the known plans for goal achievement have failed. This representation can also be used for plan simulation and testing. By propagating the effects of the plan across the network according to the links established between the nodes, the plan can be monitored for defects. For example, using Rieger's (1975:9-13) notations and definitions, a pilot's understanding of the oxygen/altitude relationship in an unpressurized aircraft could look like Figure 5. The corresponding code similar to Underwood's (1983:303) and definitions are in Figure 6.

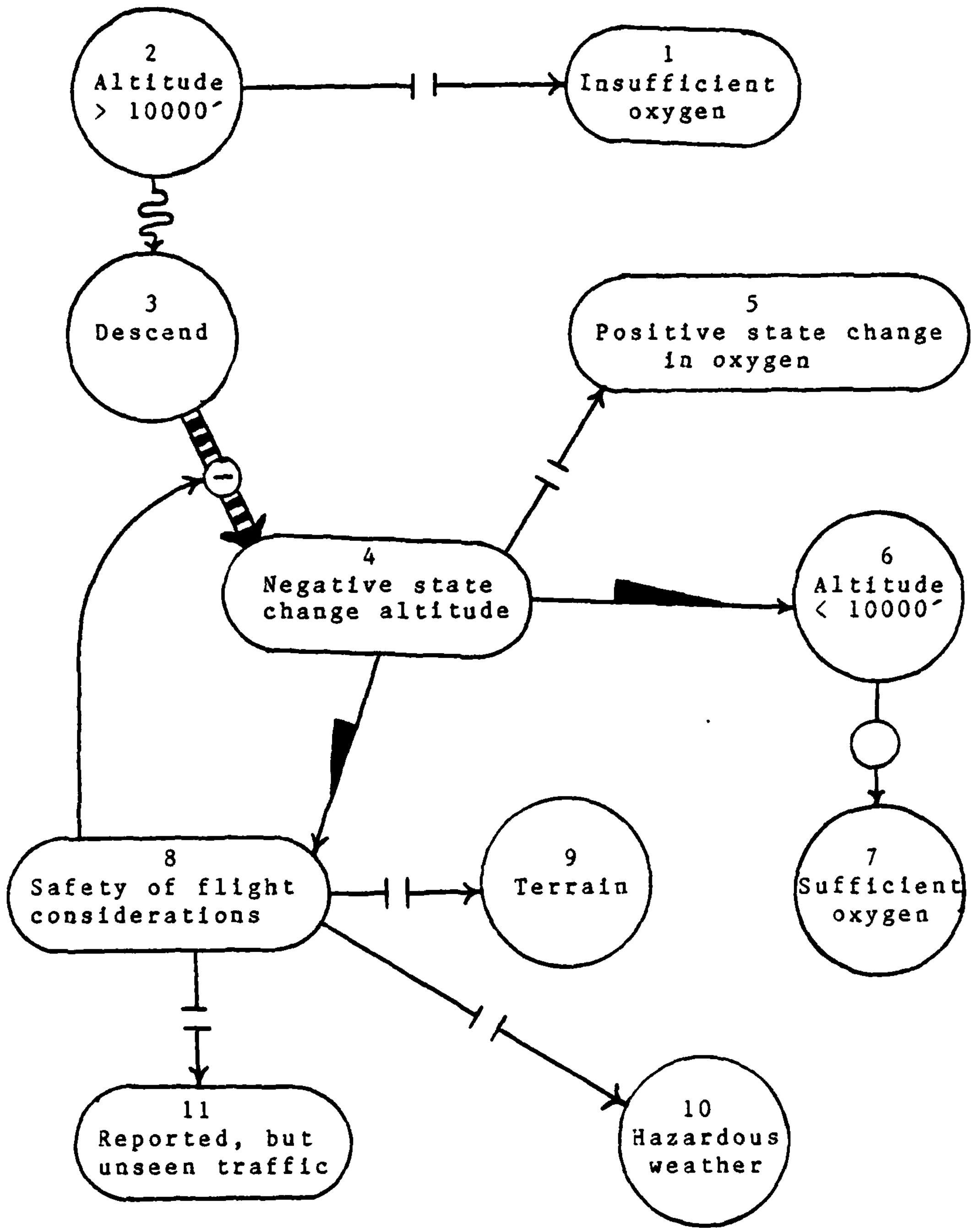


Figure 5. The CSA Representation Diagram for the Oxygen/Altitude Relationship

```
($DOMAIN
  (NAME OXYGEN_SYSTEM)

  (EVENTS
     (1 S (INSUFFICIENT OXYGEN))
     (2 S (ALTITUDE > 10000'))
     (3 A (DESCEND))
     (4 SC (ALTITUDE NEGATIVE))
     (5 SC (OXYGEN POSITIVE))
     (6 S (ALTITUDE < 10000'))
     (7 W (SUFFICIENT OXYGEN))
     (8 S (SAFETY OF FLIGHT CONSIDERATIONS))
     (9 S (TERRAIN))
     (10 S (HAZARDOUS WEATHER))
     (11 S (REPORTED, BUT UNSEEN TRAFFIC))

      (LINKS
          (S-COUPLE (2 1))
          (C-ENABLE (2 3))
          (C-CAUSE  (3 4))
          (S-COUPLE (4 5))
          (THRESH (4 6))
          (THRESH (4 8))
          (G-REAL (6 7))
          (DISABLE (8 (3 4)))
          (S-COUPLE (8 9))
          (S-COUPLE (8 10))
          (S-COUPLE (8 11))

     (PURPOSE (1 7))
     (NORMAL (6))
     (TRIGGER (2)))
```

Figure 6. The CSA Representation of the Oxygen/Altitude Relationship

The CSA representation used in Figures 2 and 3 consists of four state events and six links or relations. The four events are actions (A), states (S), statechanges (SC), and wants (W). The links include causal state coupling (S-COUPLE) where state one is synonymous with state two, threshold (THRESH) which triggers a new state, goal realization (G-REAL), continuous causality (C-CAUSE) where the action yeilds the state change, continuous enablement (C-ENABLE) where the state is requisite to the action, and disablement (DISABLE) which halts the state changes. Other events and links are used and described by Rieger (1975, 1976) and Underwood (1983).

The Plan Proposer can now use this CSA representation to help construct a plan to try to satisfy the goal "get oxygen to passengers." After the known plans to descend below 10,000 and 13,000 feet respectively were rejected, the Proposer enters the model at the state "Altitude > 10000′." This state continuously enables the action "Descend" which has the effect of decreasing altitude until the threshold of below 10,000 feet or a safety of flight restriction stops the descent. In the example of flying in an area of mountainous terrain, the high terrain will block the descent before going below 10,000 feet. Therefore, the Proposer will return the plan "fly as low as safely possible." The lowest safe altitude information would be available from the the navigation module of the Intelligent Pilot Aid.

The CSA representation above is a very basic representation and would need to include much more information, but it shows how it would be used and implemented. Other information which would increase the planner's understanding of the oxygen relationships could include actual amounts of oxygen versus altitude, specific threshholds concerning human's oxygen requirements, the effects of a true rapid decompression on time of usefull consciousnous (TUC), supplemental oxygen, partial presurization of the aircraft, and others.

Of over 50 emergency procedures prescribed in the C-130 Flight Manual, nearly two-thirds require the "turning on" or "turning off" of switches to various systems on the aircraft. It is through these switches the crew controls the electrical, fuel, hydraulic, bleed air, air conditioning, communication and navigation functions. The emergency actions to "discharge fire agent" and "shear generator shaft" are also switch activated.

During some emergencies, not enough information is initially available to construct a full plan that will get the aircraft on the ground safely. In some cases, the first plan step may be "get more information." Therefore, another task the Plan Proposer and Projector must perform is "trouble-shooting" the cause of certain observations. A "generator--ON" light calls for checking frequency, voltage and load. If they all read "0", turn off the generator, reset, turn it on and monitor. If the readings return to "0", use the generator disconnect or shut down the engine to prevent generator break-up which could cause an engine fire. There are other options, but the point is the planner has to have an "understanding" of the systems and the causal relationships between them.

The work in this area conducted by Chien seems particularly appropriate as it "deals with the development of computer-based diagnosis and design methodologies rooted in deep-level understanding models" (Waltz, 1983:27). His efforts focussed on building a system which understands a DC-10 electrical generating system at three levels, "system," "subsystem," and "component." The obvious implication is that if the planner can reference its diagnostic model of the system in trouble, determine either the cause or resultant effect, it can better form a successful plan of action.

Consider the emergency example in Appendix C of the C-130 experiencing a "Right-wing Overheat Warning Light" while flying in icing conditions. Turning off the right-wing anti-icing system allows a rapid and dangerous ice buildup on the right-wing, but turning on the anti-ice system with a known overheat condition, risks the danger of a wing fire or explosion. Knowing the details about the overheat detection system, the bleed air system, and the aircraft icing characteristics makes it easier to plan a safer course of action during this kind of goal conflict.

Qualitative Reasoning

Another approach to plan simulation, projection, and testing that a PIPA could use is qualitative reasoning, the process of drawing conclusions or inferences from possibly incomplete data, knowledge, or observations (Cross, 1983:55).

A pilot planning his course of action following an inflight emergency frequently makes decisions despite uncertainty about certain aspects of his aircraft's capabilities. He can't be sure how much performance degradation the ice accumulation on his wings is causing (failed wing anti-ice system) to accurately calculate if he can make his intended destination. But he must still decide something to try to effect a safe termination to the problem.

In the enroute air traffic control domain, Cross (1983) has implemented an expert system which can justify heuristically generated plans by applying qualitative reasoning to aircraft performance equations. He represents the equations in a semantic network with the nodes representing the variables and the links representing the dependent variable influences. By constructing well-founded explanations for its planning actions based on the evaluation of detailed equations, the system can generate understandable explanantions for the human controllers.

The Projector could use this qualitative reasoning structure to help justify a plan to the pilot by describing in "naive physics" terms, the effects of the plan. The Projector could project the plan by propagating the constraints in a semantic network model of the equations of the actions of an aircraft. This type of reasoning would be especially appropriate for planning maximum performance take-offs, landings, and obstacle clearance decisions where safety goals and mission goals frequently conflict. For example, is the aircraft light enough to make a short-field landing or should the crew dump fuel, a resource wasting plan. The loss of an engine during a heavy-weight take-off after refusal speed has been called, will the aircraft really make it off the runway before the overrun?

The United States Navy is currently using STEAMER (Forbus, 1981) to give student sailors an understanding of a ship's steampower system. STEAMER is a dynamic learning environment embodying the use of colorful graphical displays, numerical simulation, and emphasis on understanding the rationale behind the required procedures (Milne, 1984). The effects of a plan "open valve 1" are shown on a schematic of the valves, pipes, gauges, boilers, etc., involved in the system. It basically operates like a simulator, testing the plans proposed by the students, allowing them to decide if the plan they have chosen was successful or not.

Some of these same capabilities could be utilized by the inflight emergency domain planner during its projection of proposed plans. By observing the plans effects on the model, an opinion of the plan can be generated. The qualitative reasoning used in STEAMER involves the use of several feedback mechanisms, as the effects of a plan propagate through the model.

The Executor

The Executor's implementation was distributed among the whole of the program in the form of messages and questions to the pilot. Since it controlled no functional system, the only module created for it was the one used to receive questions and return answers about the plans, goals, and observations.

Rapid Decompression Implemented

The Rapid Decompression emergency shows the step by step planning process used in the implementation.

The planner begins by asking the user whether an observation, the IF part of the rule, is true or not. If true, it adds the THEN part or goal to the goal list. The observation (cabin altitude greater than 10000 feet) is true and the goal (get oxygen to crew and passengers) is detected and passed to the Proposer. The goal is actually written as two goals (get oxygen to crew) and (get oxygen to passengers).

The Plan Proposer searches its list of frames looking for the frames with the specified goals using the function

(slot-val plan 'goal).

"Slot-val" retrieves the "plan" associated with "goal". The plan (don oxygen masks) is retrieved for the crew goal, from the slot called "check-list plan." The slot "preconditions" contains two items, (crew have masks) and (oxygen in system), which are compared with the "world-state" list, to decide if the preconditions are satisfied. They are satisfied in this case.

At this point a small simulation checks to verify that enough oxygen is available for the suspected duration of the emergency based on the expected flight plan time. A simple calculation of dividing (litres available) by (rate of use per hour) would equal (hours of oxygen available). This figure would be compared to the amount of calculated time it will take to get below 10,000 feet, a figure obtainable from the navigation module or the navigator.

The Projector passes this plan (don oxygen masks) to the Executor which in turn relays it to the pilot.

The Proposer uses the same steps to find the plan (descend below 10000 feet) for the passengers and lets the Projector look it over. Seeing no preconditions, the Projector simulates a simulation which has the effect of comparing the plan altitude of 10,000 feet to the minimum

Emergency Safe Altitude (ESA) which is set at 15,000 feet because of the mountains. The comparison fails and the plan is rejected and added to the failed-plan list for the reason "illegal descent---too low."

The Proposer sends the plan (descend below 13,000 feet) and it too is rejected for the same reason. Intelligent plan selection wasn't incorporated in this implementation, but the 13,000 foot plan would have been suggested anyway unless the Projector said it needed an altitude greater than 15,000 feet.

The Proposer suggests its last plan of (use smoke masks) with the precondition there be (one per passenger). There are only five smoke masks onboard, so this plan is rejected for the lack of a satisfied precondition. However, before the rejection, the Goal Detector spots this and detects the goal (get more smoke masks). The Proposer finds (call life support) with the preconditions (be on the ground) and (be near life support office). Basically, the plan to use smoke masks requires flying back to base to get more masks. Because it is not a practical plan for this situation, it is rejected. Actually, when the first precondition (one per passenger) was found, the value in the note-slot indicated that this precondition was unalterable during the cruise portion of the flight.

After all known plans were tried unsuccessfully, the goal was abandoned. The plan "descend as low as safely possible" could have been included in the list of plans, the intent here was to investigate how the planner reason about the domain and come up with a probable plan on its own. This is the place where qualitative reasoning and CSA representation could apply. If the Proposer had been able to make its own plan, the following describes the last steps.

The Proposer finds the plan using the process described earlier in this chapter under the CSA discussion (descend as low as safely possible) and has the Projector test it. The Projector gets from the Navigation Module the information that as-low-as-safely-possible is equal to 15,000 feet at this time. The test is successful and the plan is passed to the pilot via the Executor. The pilot may ask the Executor how long can the passengers can live or be okay and it should give the reply it recieves from the "areospace-physiology-module."

During the execution of the plan to descend, the Goal Detector is waching to see if it is safe to descend further, because the goal to "get oxygen to passengers" was only partially satisfied. As soon as the airplane descends through the 10,000 foot level, the oxygen rule may be placed back in the Detector's actively monitored list.

The next chapter discusses the appropriateness and implementation of this planning approach in the flight domain. Recommendations are given for future research.

VI. Analysis, Conclusions and Recommendations

This final chapter provides an analysis of why Wilensky's planning approach is appropriate for the inflight emergency domain and the problems associated with multple goal conflicts in this dynamic environment. It also discusses implementation results and proposes some enhancements to the design. Conclusions from this study and recommendations for further research complete the chapter.

Analysis

Wilensky's planning approach is appropriate for the inflight emergency domain for several reasons. The planning protocals used by pilots evidenced in Appendix C are compatible and consistent with the protocals used in this planning structure. Pilots will detect a goal, recall the approriate plan, verify that it is sufficient for the situation, and implement it. The meta-planning process for guiding the protocals is implicit in the structure of the Planner for an Intelligent Pilot Aid (PIPA).

Comprised of four components, the PIPA will watch for the items it is tasked to watch during this particular phase of flight or mission, similar to the pilot or flight engineer scanning the instruments and indicator lights. When it finds something of interest such as a warning light or an overtemp, it simultaneously associates a goal with that observation and a proposes a plan to achieve the goal.

The proposed plan is tested in a hypothetical model of the current situation. If the plan is successful it is passed along to the pilot. If the plan fails because of a conflict with another goal, then PIPA detects the goal of trying to resolve the goal conflict which just arrived. Just as a pilot tries to solve his own problems, this planner handles planning goals with the same structure it uses to handle domain goals.

This planner deals with goal conflicts using essentially two methods. A conflict exists when the fulfillment of one goal interferes with the fulfillment of another goal. The planner either tries to find new plans which will achieve the goals or it will try to change the circumstances under which the conflict revolves. The planner upon detecting a conflict will check for a plan that is known to be used in this situation, such as "donning an oxygen mask" is the known plan used to resolve the conflict between the goal "remaining functionally alive (as opposed to becoming hypoxic or passing out)" and the goal "work in an oxygen deficient environment." If the plan does not work, another one, if available, is selected.

After trying unsuccessfully all the appropriate known plans for a goal, the planner tries to build one of its own by reasoning about its model of the circumstances. If this also fails, the planner then attempts to change the circumstances surrounding the goal situation.

This planning approach is appropriate for handling conflicting goals because unlike other planning structures designed to handle an only a priori specified goal, it has the structure to handle multiple goals. It can also detect its own goals, determine the priorities of conflicting goals, and abandon goals which are not possible to achieve.

Appropriate handling of goal conflicts does not imply that the planner always satisfies every goal. In many cases, goals will have to be abandoned. Reasons for abandonment include low priority, lack of available resources, or lack of other satisfied preconditions.

The goal of the implementation portion of the research was not to implement a complete planner, but to determine the adaquecy of current AI programming techiques to build the planner. Current programming capabilities are sufficient to build prototypes which will allow further study into ways to improve the planning effectiveness.

The planner consists of four components, the Goal Detector, the Plan Proposer, the Plan Projector, and the Executor. A forward-chaining rule-based production system was implemented for the Goal Detector and a frame based system represented the other components. An implementation of a simulation mechanism was proposed, because there currently does not exist a sufficient simulation mechanism which can model the more complex processes found in this domain.

Wilensky (1983a) implemented PANDORA in a frame-like structure attempting to use his model to propose plans, simulate probable futures, and detect goals. He found, though, as implemented, the model "does not possess the capability to deal with the more complex aspects of planning that we initially claimed were central to processing complex situations" (Wilensky, 1983a:26).

To achieve this capability of complex planning, this research suggests that Rieger's Commonsense Algorithms (CSA) be used to represent models of aircraft and aerospace systems for plan creation, reasoning, and simulation. An example in chapter five shows how a CSA would give the planner some reasoning capability to be able to propose plans. The same structure could with slight modification, be used to conduct the simulations for those particular goal planning situations.

For certain applications such as critical aircraft performance calculations for obstacle clearance and maximum effort landings, a model-based qualitative reasoning approach as proposed by Cross (1983) would provide an important simulation and plan justification role in the planner.

Certain aspects of the flight domain suggest that the planning structure required for this domain may be less complex than the one required for the mundane activities domain. The numerous flight regulations help constrain the number of planning options because in lieu of a regulation the planner would need to try to reason or find a cause for proposing a particular plan. Also, the actors (pilots) in this domain are highly trained and experienced in the flying environment, suggesting that their interpretations and planning actions are fairly uniform when compared with the reactions likely to be found in average people in everyday situations. The plans the planner will need to produce can act more as memory joggers rather than as original material requiring lots of detail to explain it to the pilot.

Conclusions

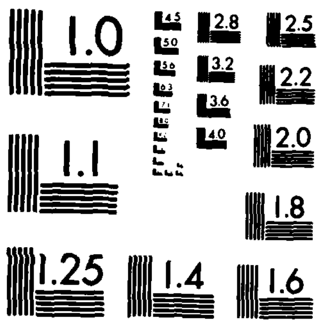
The inflight emergency domain is an extremely complex and challenging domain *requiring* of its planner large amounts of world and flight knowledge combined with common sense and expert reasoning. Even specially selected, highly trained professional pilots make occasional errors in this environment. Part of the intent of this research is to explore areas where machines might be able to help or aid pilots in these situations.

Like the domain it is tasked to understand, the planning structure for inflight emergencies is also extremely complex, requiring structures to account for every meta-planning option, whether it be to replan, change circumstances, simulate a plan, propose new plans, or abandon an impossible goal. While some problems and

AD-A151 869 AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING D L KNODE DEC 84 2/2

UNCLASSIFIED AFIT/GCS/ENG/84D-15 F/G 6/4 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

difficulties may hinder immediate effective implementation, slowly but surely new concepts and approaches will diminish these obstacles, allowing a machine to some day actually become an Intelligent Pilot Aid.

Recommendations

This research investigated Wilensky's theory of planning. He has also proposed a theory of understanding which ties in closely with planning. These capabilities should be combined into one pilot aid.

For a planner to be effective in the emergency domain, it must already be proficient in the "normal" domain. For actual initial implementation the normal domain will present enough challenges until more is learned about model-based reasoning. The normal C-130 electrical system alone has five AC generators, a battery, two transformer-rectifiers, and numerous busses (both AC and DC), etc., and represents a reasonably difficult starting point.

Rieger's commonsense algorithms (CSA) would be an likely representation to use for testing system modelling, as would Forbus' representation used in STEAMER, a tool already used to teach US Navy sailors. These approaches could be investigated separately or compared to each other.

With the increasing complexity of aircraft systems, malfunction diagnosis also becomes more complex. If pilots cannot diagnose the problem correctly, it is unlikely they will find the correct solution to the planning problem either. A complete model of a system using CSAs should allow the ability to simulate malfunctions or at least play what-if questions with it. Certainly the first generation of cockpit aids will be just that, aids. The pilot can do the correct planning as long as time and valid information are available.

Goal priorities can not always be explicitly established. The approach to prioritize goals in this planner was partially explicit and partially simulate the plans and select the best option. Since simulation is time consuming and perhaps not yet mature enough, more research into ways to prioritize goals in a dynamic environment is necessary.

Appendix A:

Flight Domain Regulation Excerpts

The following excerpts help define the guidelines the regulations provide for crewmembers in the operational flying environment. They also shed some light on the spirit of the planning process, that is, despite all the millions of dollars for design and testing and all the lives lost in the process of compiling the manual, whenever a crew goes to fly they must ultimately rely on their own skill and merits.

KNOW IT BEFORE YOU GO,
OR YOU WILL GO BEFORE YOU KNOW IT

SCOPE

This manual contains the necessary information for safe and efficient operation of your aircraft. These instructions provide you with a general knowledge of the aircraft and its characteristics and specific normal and emergency operating procedures. Your experience is recognized; therefore, basic flight principles are avoided. Instructions in this manual are prepared to be understandable by the least experienced crew that can be expected to operate the aircraft. This manual provides the best possible operating instructions under most circumstances, but it is not a substitute for sound judgement. Multiple emergencies, adverse weather, terrain, etc. may require modification of these procedures.

PERMISSIBLE OPERATIONS

The flight manual takes a "positive approach" and normally states only what you can do. Unusual operations or configurations are prohibited unless specifically covered herein. Clearance from the using command must be obtained before any questionable operation, which is not specifically permitted in this manual, is attempted.

(TO 1C-130A-1 : ii)

The following excerpt is the introduction to the Emergency Procedure chapter of the C-130A Flight Manual. It further clarifies the "planning environment."

This section contains the procedures to be used in coping with the various emergencies that may be encountered. A thorough knowledge of these procedures will enable crew members to perform their emergency duties in an orderly manner, and to determine the seriousness of the emergency. This will permit early planning for a bailout or forced landing and will greatly increase the crew's chances for survival. The procedures consist of items classified as critical or noncritical. The critical items are actions that must be performed immediately to avoid aggravating the emergency and causing injury or damage. Critical items are presented in bold face type and must be committed to memory. Noncritical items are actions that contribute to an orderly sequence of events and will be performed with reference to the appropriate checklist before performing them or afterward as a cleanup reference. After determining that an emergency exists, the pilot should immediately establish communication with a ground station. The ground stationshould be given a complete description of the emergency, the action taken, and an accurate position report. The ground station should be further notified of any changes or developments in the emergency, so that the station can alert Aerospace Rescue and Recovery Service (ARRS) or other agencies to standby if necessary, In the checklist presented, the codes P, CP, E, N, and LM stand for pilot, copilot, flight engineer, and loadmaster. This presentation does not preclude the pilot from redelegating the duties at crew briefing. Never initiate bold face procedure before command of the pilot. The pilot will command initiation by calling for the procedure desired, but need not call out each step. The affected crew members will accomplish the required steps in accordance with the appropriate checklist. The engineer will monitor all engine shutdown steps and other coordinated emergency procedures. Regardless of specific emergency *encountered:*

a. Maintain airplane control.

b. Analyze the situation.

c. Take coordinated corrective action.

(TO 1C-130A-1, Change 3 : 3-3)

The following is the Rapid Decompression Emergency Procedure from chapter three of the TO 1C-130-1, page 3-40.

RAPID DECOMPRESSION

Sudden and uncontrollable loss of cabin pressure is known as rapid decompression. This may result from losing a nonstructural member, such as a door or window, or from a rupture in the fuselage. If a rapid decompression occurs, proceed as follows:

1. Oxygen - As required. (P)

Pilot will direct the crew to go on 100% oxygen as required.

The flight engineer should make an inspection of the fuselage during descent (using a walk-around oxygen bottle, is required, or a parachute if a restraint is not available) to determine what caused the decompression and the extent of any damage. With no structural damage, descent airspeed may be increased not to exceed maximum speeds, as shown in Section V. With structural damage, the flight will be completed at a safe speed as determined by the pilot. The flap configuration for landing will depend on the type of structural damage.

If descent is required, continue as follows:

2. Throttles - FLIGHT IDLE.

3. Descent - As required.

****
* CAUTION *
****

With certain types of structural damage, changing the center or lift with the flaps may induce further damage. Careful consideration should be given to changing airplane configuration.

Appendix B:

Oxygen Requirements,

An Example of Domain Knowledge

The following text is the guidance contained in Air Force Regulation 60-16 (Change 1), 18 June 1981, pertaining to Oxygen Requirements.

*6-6. Oxygen Requirements. Where the cabin altitude exceeds 10,000 feet, each occupant of an Air Force aircraft must use supplemental oxygen except as noted below.

a. Unpressurized Aircraft:

(1) If the minimum enroute altitude or an ATC clearance requires flight above 10,000 feet MSL in an unpressurized aircraft, the pilot at the controls must use oxygen.

(2) If oxygen is not available to other occupants, flight between 10 and 13,000 feet MSL must not be longer than 3 hours, and flight above 13,000 feet MSL is not authorized.

(3) If all occupants are equipped with oxygen, flights may be conducted up to flight level 250.

b. Pressurized Aircraft. When an aircraft is flown over 10,000 feet MSL, but its cabin altitude is maintained at 10,000 feet or less, oxygen equipment is used as specified in table 6-1 (next page).

(1) A MAJCOM may establish more restrictive procedures

for using oxygen during ground or flight operation of tactical aircraft or jet trainers if required.

(2) Enough oxygen must be onboard an aircraft before its takeoff to fly the planned mission.

(3) If the aircraft loses pressure, it will descend immediately to a point where a cabin altitude can be maintained at or below flight level 250, unless the occupants are wearing a functional pressure suit.

(4) If the aircraft loses pressure, and any occupant lacks functional oxygen equipment, descend to maintain a cabin altitude of 10,000 feet or less and comply with a above.

NOTE: If an occupant appears to be suffering decompression sickness, the pilot will descend as soon as practical, and land at the nearest suitable installation where medical assistance can be obtained. Before the person affected may continue the flight, he or she must have a consultation with a flight surgeon or flight medical officer or a civilian aeromedical examiner.

TABLE 6-1. Oxygen Requirement for Pressurized Aircraft

Ambient Altitude in Feet	Pilot	Pilot*	Occupants
10,000 ft through FL 250	R	R	NA
Above FL 250 through FL 350	I	R	R
Above FL 350 through FL 410	I	I	R
or	O	R	R
Above FL 410 through FL 450	O	I	R
Above FL 450 through FL 500	O	I	I
Above FL 500	P	P	P

Legend:
R--Oxygen must be readily available. A functioning system and mask must be located within arm's reach, and the regulator set to 100 percent and ON, if the system contains an operator adjustable regulator.
I--Oxygen must be immediately available. Helmets must be worn with an oxygen mask attached to one side, or an approved quick-donning or sweep-on mask properly adjusted and positioned for immediate use. Set oxygen regulator to 100 per cent and ON.
O--Oxygen must be used.
P--Pressure suit must be worn.
*These requirements also apply to non-pilot crewmembers occupying crew positions with direct access to flight controls.

Appendix C:

Emergency Procedure Transcripts

for Five Selected Emergencies

This appendix contains the emergency procedure actions which three highly qualified pilots said they would take in response to the particular emergency situation. These pilots represent over 10,000 flying hours, mostly in the C-130.

Note the variations in some of these plans. Because most of these situations are not specifically addressed in the Emergency Procedures chapter of the Flight Manual, the pilot must use his best judgement and experience to help determine his course of action. Also note the protocals the pilots use.

Situation 1: Your aircraft experiences a rapid decompression while flying at FL250 (25,000 feet above sea level). You have 25 passengers on board.

Pilot 1:

1. I´d don my oxygen mask, calling for my crew to do likewise and check-in on interphone when done.

2. Declare an emergency and request the Emergency Safe Altitude (ESA) from the navigator and begin a descent to that altitude.

3. Check with the loadmaster for the cause of the

depressurization and the status of the passengers.

4. Have the loadmaster start using the smoke masks and walk-around oxygen bottles to revive any of the passengers needing some oxygen and quickly instruct them to help each other until we reached a lower altuitude.

5. Start flying toward terrain which will allow a further descent and subsequent landing.

6. If not a viable option, would investigate a temporary repair of the pressurization problem.

Pilot 2:

1. Descend immediately to lowest safe altitude, declare emergency, and accomplish checklist.

2. Send radio operator to the back to assist the loadmaster and pararescue specialists buddy breath with the passengers while I and the navigator obtain the best heading for lower terrain (meanwhile the engineer attempt troubleshoot of the depressurization to attempt repressurization).

Pilot 3:

1. Direct crew to don oxygen masks, begin descent to ESA, and declare an emergency.

2. Obtain status of passengers from the loadmaster and direct him to assist them as necessary.

3. Coordinate with the navigator the best route to

the lowest altitude, at least below 10,000 feet, with suitable landing destination.

4. Troubleshoot problem and attempt repressurization if feasible.

<u>Situation</u> <u>2</u>: Wing overheat light illuminates while in icing conditions.

Pilot 1:

1. I'd go through the wing overheat procedures and attempt to alleviate the problem before it gets worse.

2. Divert from my current flight plan to avoid the icing as much as possible.

3. Avoid areas of visible moisture and seek an altitude above or below the worst of it.

Pilot 2:

1. Follow overheat procedures, change altitude or temperature.

2. If icing becomes unbearable, open bleed-air valve furtherest from the overheat section and await Situation 5. P.S. I had a similar scenario and the airplane remained controllable without restoring bleed-air. Dash one procedures pretty much cover this situation.

Pilot 3:

1. Follow wing overheat procedures as outlined in checklist.

2. Inform Air Traffic Control (ATC) of situation and request permission to manuever to avoid icing and visible moisture. Probably would have to descend if much airspeed or power was lost.

3. If too much ice was forming on wing, would attempt to melt some of it by opening a distant bleed-air valve to provide a minimum blast of hot air to that overheated section.

4. Plan to land as soon as possible.

Situation 3: During a heavyweight take-off, you experience the loss of an engine (flameout) after refusal speed and a second loss (overheat) is pending. Your departure path overflies a city.

Pilot 1:

1. Declare an emergency, maximize airspeed, and reduce drag.

2. Continue using "bad" engine while trying to align self for fuel dumping. Begin dumping to below 100,000 pounds.

3. Keep bleeds closed, watch my turns and angle of bank, and when light enough, go in for the landing.

Pilot 2:

1. Keep second engine going.

2. Dump fuel.

3. Begin teardrop back to runway for landing.

4. Attempt restart on first engine, depending on how it looks (no smoke, missing panels, etc.).

Pilot 3:

1. Continue departure with second engine running.

2. Declare emergency while setting up to dump fuel, avoiding the city if possible.

3. Begin the dumping process until under 110,000 pounds, planning the dump to be in a position to make an immediate landing, before I lose that second engine.

Situation 4: You are flying above a 10,000 feet thick cloud layer when you lose your attitude indicator. There are no VFR destinations within fuel range.

Pilot 1:

1. First option. Get a chase C-130 to fly formation on until below the clouds for the VFR landing.

2. Else, attempt to regain attitude indicator. Use turn and slip indicator to fly down through the clouds.

3. Choose airport with flattest terrain and longest runway, using a long, straight-in approach, configuring before going IFR.

Pilots 2 & 3:

1. If chase plane for a formation approach is unavailable, I'd try a long, straight-in, gyro-out PAR approach, configuring before entering the clouds, using needle, ball, and airspeed.

Situation 5: Following engine shutdown for fire, it spreads to the wing. You descend and accelerate. It goes out at 4000 feet and 350 knots. Just when things calm down a bit, the fire reignites.

Pilot 1:

1. Attempt to put out the fire. Consider shutting down the other engine on that wing, and getting configured for an immediate landing.

2. Order a bailout if parachutes are on board and fire persists.

3. Else set up for landing, hoping for a close-by airport, opting for a straight public road. Have the overhead escape hatch open and the paratroop doors open and locked.

4. Land and immediately evacuate the burning aircraft.

Pilot 2:

1. Bailout if chutes are onboard.

2. Failing that, go fast as possible to landing sight for ASAP landing. Danger is that the aircraft can go uncontrollable very quickly.

Pilot 3:

1. Order crew to bailout if chutes on board and fire continues to rage. Attempt to stay with plane until everyone else out and then either emergency land it or bailout myself.

2. If no chutes, fly to nearest landing site (runway, road, field, etc.) and put it down and evacuating it as quickly as possible. Request fire fighting squads to be standing by.

Bibliography

Air Force Safety Center. Norton Air Force Base, California, telephone conversation, (January 1984).

Abelson, R.P. "Concepts for Representing Mundane Reality in Plans," In Bobrow, D., and A. Collins(eds). Representation and Understanding: Studies in Cognitive Science. New York: Academic Press, 1975.

Air Force Studies Board. Automation in Combat Aircraft. National Research Council, National Academy Press, Washington, D.C., 1982.

Anderson, Bruce M, and others. "Intelligent Automation of Emergency Procedures in Advanced Fighter Aircraft," Proceedings of the First Conference on Artificial Intelligence Applications, Denver, Colorado: to be published (December 5-7, 1984).

Bobrow, Daniel G. and others. "GUS, A Frame-Driven Dialog System," Artificial Intelligence 8. North-Holland Publishing Company, 1977.

Barr, Avron and Edward A. Feigenbaum. The Handbook of Artificial Intelligence. Los Altos, California: William Kaufmann, Inc., 1981.

Barr, A. "Meta-knowledge and Memory." Working Paper. Stanford University Heuristic Programming Project HPP-77-37, 1977.

Carbonell, J. G. "Subjective Understanding: Computer Models of Belief Systems." Research Report #150. Yale University Department of Computer Science, 1979.

Charniak, E., Christopher K. Riesbeck, and Drew V. McDermott. Artificial Intelligence Programming. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc., 1980.

Charniak, E. "A Framed PAINTING: The Representation of a Common Sense Knowledge Fragment," Cognitive Science, Vol 1, No. 4, 355, 1977.

Cohen, Paul R. and Edward A. Feigenbaum. The Handbook of Artificial Intelligence, Vol. 3. Los Altos, California: William Kaufmann, Inc., 1982.

Cross, Stephen E. "Towards a Flight Domain Expert System Architecture," Proceedings of the National Aerospace and Electronics Conference: 784-788 (May 1984).

Cross, Stephen E. Qualitative Reasoning in an Expert System Framework, PhD Dissertation. University of Illinois, Urbana-Champaign, Illinois, May 1983.

Davis, Randall. "Meta Rules: Reasoning About Control," Artificial Intelligence, Volume 15:279-322, 1980.

Davis, R., and B.G. Buchanan. "Meta-level Knowledge: Overview and Applications," Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, (1977).

Deering, M. and others. "Using the PEARL AI Package," No. UCB/ERL/M82/19, Berkeley Electronic Research Laboratory Memorandum, Berkeley, California, (1982).

Defense Advanced Research Projects Agency. Strategic Computing. DARPA, (October 1983).

Department of the Air Force. General Flight Rules. AFR 60-16. Washington: HQ USAF, 5 December 1980.

Faletti, J. "PANDORA---A Program for Doing Commonsense Planning in Complex Situations," Proceedings of the Second Annual National Conference on Artificial Intelligence. Pittsburgh, Pennsylvania, (1982).

Fikes, R.E. and N.J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, Vol. 2, No. 3-4, 189-208, 1971.

Forbus, K. and A. Stevens. Using Qualitative Simulation to Generate Explanations. Report No. 4490, Bolt, Beranek, and Newman, Inc., 1981.

Hammer, John M. "An Intelligent Flight Management Aid for Procedure Execution," Unpublished Paper. Center for Man-Machine Systems Reasearch, Georgia Institute of Technology, Atlanta, (1983).

Hayes-Roth, Frederick, Donald A. Waterman, and Douglas B. Lenat. Building Expert Systems. Reading, Massachusetts: Addison-Wesley Publishing, 1983.

Hayes-Roth, Barbara and others. Human Planning Processes. R-2570-ONR, Rand Corporation, Santa Monica, California, December 1980, (AD-A095107).

Hayes-Roth, B., and Hayes-Roth, R. "A Cognitive Model of Planning," Cognitive Science. Vol. 3, No. 4. (1979).

McDermott, D.V. "Flexibility and efficiency in a computer program for designing circuits," TR-402, Massachusetts Institute of Technology Artificial Intelligence Laboratory, (1977).

Millar, Robert J. An Artificial Intelligence Approach for Planning Air Launched Cruise Missile Missions. MS Thesis, to be published, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984.

Milne, Robert. Lecture materials distributed in EE7.49, Expert System Building. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, April 1984.

Minsky, M. "A Framework for Representing Knowledge," in P.H. Winston (ed). The Psychology of Computer Vision. New York: McGraw-Hill, 1975.

Newell, A and H. A. Simon. Human Problem Solving. Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

Rieger, Chuck. The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief, and Contextual Language COmprehension. Technical report 373, University of Maryland, College Park, Maryland, (May 1975).

Roberts, R.B. and I.P. Goldstein. The FRL Manual. Memo 409, MIT AI Laboratory, Cambridge, Massechusettes, (1977).

Rouse, S.H. and others. "Design and Evaluation of an Onboard Computer-based Information System for Aircraft," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, No 4. (July/August 1982).

Sacerdoti, E.D. A Structure for Plans and Behavior. Elsevier North-Holland, Amsterdam, 1977.

Schank, R., and R. Abelson. Scripts, Plans, Goals, and Understanding. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1977.

Schira, J.A. Jr., J.R. Jurgensen, B. Deer, and J.L. Girard. Expert Systems Combat Aid to Pilots, AFWAL-TR-84-1109. Dayton, Ohio: SYSTRAN Corporation, July 1984.

Stefik, J.R. "Planning with Constraints," Work Paper. Stanford Heuristics Programming Project HPP-80-12. (1980).

Sussman, G.J. A Computer Model of Skill Acquisition. American Elsevier, New York, 1975.

Tate, A. "Interacting Goals and Their Use," Proceedings of the Fourth International Joint Conference on Artificial Intelligence. Tbilisi, Georgia, USSR, (1975).

Underwood, W. E. "A CSA Model-Based Nuclear Power Plant Consultant," Proceedings of the National Conference on Artificial Intelligence. Carnegie-Mellon University, University of Pittsburg, Pennsylvania. August 1982.

Waltz, D.L., G. DeJong and R.T. Chien. An Expert Distributed Robotics System with Comprehension and Learning Abilities in the Aircraft Flight Domain, 1 Jan 82--31 Dec 82. Contract F49620-82-K-0009. Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign IL, Feb 83 (AD-A127739).

Warren, D.H.D. "WARPLAN: A System for Generating Plans," Department of Computational Logic, Memo No. 76, University of Edinburgh, Edinburgh, Scotland, (1974).

Wiener, E.L., and R.E. Curry. "Flight-deck Automation: Promises and Problems," Ergonomics, Vol. 23, No. 10, (1980).

Wilensky, Robert. LISPcraft. New York: W.W. Norton and Company, 1984.

Wilensky, R. Planning and Understanding: A Conceptual Approach to Human Reasoning. Reading, Massachusetts: Addison-Wesley Publishing Company, 1983a.

Wilensky, R. and others. "PEARL---A Package for Efficient Access to Representations in LISP," Working Paper. Computer Science Division, University of California, Berkeley, (1983b).

Wilkins, David. Domain Independent Planning: Representation and Plan Generation. Contract AFOSR-TR-82-1074, SRI International, Menlo Park, California, (August 1982) (AD A123169).

Winston, Patrick Henry and Berthold Klaus Paul Horn. LISP. Reading, Massachusetts: Addison-Wesley, 1981.

Winston, Patrick Henry. Artificial Intelligence. Reading, Massachusetts: Addison-Wesley Publishing Company, 1977.

VITA

Captain David L. Knode was born on 24 January 1952 in Riceville, Iowa. He graduated from the Riceville Community High School in 1970, from Waldorf College, Forest City, Iowa, in May of 1972 with an Associate of Arts in Mathematics, and finally from Iowa State University, Ames, Iowa, in May 1974 with a Bachelor of Science degree in Computer Science. Receiving his USAF commission also in May 1974, Captain Knode entered Undergraduate Pilot Training at Laughlin AFB, Texas in January 1975. He served as a Tactical Airlift Pilot in the C-130 at Yokota AB, Japan for two years and then as an HC-130 Rescue Aircraft Commander flying out of RAF Woodbridge, United Kingdom until entering the Air Force Institute of Technology, School of Engineering, in May 1983.

Permanent address: RR 1, Box 102
LeRoy, MN 55951

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GCS/ENG/84D-15

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/EN

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Information Proc. Tech. Br.

8b. OFFICE SYMBOL *(If applicable)*
AFWAL/AAAT

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State and ZIP Code)*
Wright-Patterson AFB, OH 45433

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
David L. Knode, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ______ TO ______

14. DATE OF REPORT *(Yr., Mo., Day)*
1984 December

15. PAGE COUNT
118

16. SUPPLEMENTARY NOTATION
[illegible handwriting]

17. COSATI CODES

FIELD	GROUP	SUB. GR.
06	04	
09	02	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Artificial Intelligence, Pilots, Planning, Reasoning, Emergencies, Theses

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: AN APPROACH TO PLANNING IN THE INFLIGHT EMERGENCY DOMAIN

Thesis Advisor: Stephen E. Cross, Captain, USAF

IAW AFR 190-17.
Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Stephen E. Cross, Captain, USAF

22b. TELEPHONE NUMBER *(Include Area Code)*
(513) 255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Abstract

A planner must understand its domain and be able to effectively reason about interacting goals competing for satisfaction in its environment. Current artificial intelligence (AI) planning structures are inadequate for expert and commonsense reasoning in the dynamic aircraft inflight emergency domain. These current planners are inadaquate because they are not designed to manipulate multiple goals in an unpredictable environment, nor are they equipped to simulate dynamic, time dependent processes. Conflicting goals, i.e., when the realization of one goal interferes with the realization of another goal, poses particularly frustrating problems for typical planners.

This research discusses a new planning approach for the inflight emergency domain based on Wilensky's planning theory for the everyday activities domain. Like the everyday activities domain, the flight domain requires a large pool of world and common sense information. It also requires flight domain information and knowledge of the goals and plans of its pilots and other aircrew members.

The planner for an intelligent pilot aid (PIPA) divides planning into four activities, 1) goal detection, 2) plan proposition, 3) plan projection, and 4) execution; composed into four components with similar names. Goals are associated with the observations which trigger them, and plans are associated with the goals to which they apply. Proposed plans are simulated in a hypothetical model of current states and are watched for weaknesses, overlap, or conflict. Overlapping plan steps are appropriately combined, while conflicts direct the PIPA to either prospose and test new plans or have the PIPA attempt to change the circumstances surrounding the conflict. When a conflict cannot be resolved, the less important goal is abandoned.

Implementation of the PIPA was partially completed, but more research in the areas of simulation and model-based reasoning is required. Qualitative reasoning and common sense algorithm (CSA) representation are proposed as possible solutions toward this end.

END

FILMED

4-85

DTIC