Enroute Flight Planning: Evaluating Design Concepts for the Development of Cooperative Problem-Solving Concepts

Philip J. Smith
Cognitive Systems Engineering Laboratory

and

C. Elaine McCoy
Aviation Institute
University of Nebraska

Ames Research Center
Moffett Field, California 94035

Contract No. NCC 2-615
Progress Report
RF Project No. 767591/722399

December 1991

Introduction

The purpose of this report is to summarize the activities completed as part of this project during the last year. A more complete report will be provided upon the completion of the experiment we are conducting (scheduled to be done on March 30, 1992).

Background

Broadly speaking, our research has two goals, one applied and one more basic in nature. Specifically, our goals have been to:

1. Develop design concepts to support the task of enroute flight planning;
2. Within this applied context, to explore and evaluate general design concepts and principles to guide the development of cooperative problem-solving systems.

Specific Research Questions

Our goal is to develop a detailed model of the cognitive processes involved in flight planning. Included in this model will be the identification of individual differences (i.e., we may end up with several models to account for different subjects' behaviors). Of particular interest will be differences between pilots and dispatchers. Also included in this model will be a description of how different design features influence planning processes. Specifically, the effects of different system designs on the exploration and evaluation of alternative plans will be studied.

Our primary focus in this study is the effect on performance of tools that support planning at different levels of abstraction. Secondary issues that are also being studied include the use of different interface design features that we have incorporated (such as the graphical interface we have developed and the effects of such a graphical interface on reasoning about uncertainty).

Broader Goals

By studying the effects that alternative design concepts have on flight planning activities, and by developing cognitive models to account for these effects, we hope to produce results that will change the behaviors of future system designers and, consequently, change the ultimate designs of future aviation systems. We believe this perspective is an important one. It raises the question:

> What type of studies and results will influence the behaviors of future system designers?

We further believe that the approach taken in current study is a model for producing such an impact.

In particular:

1. By creating a functional prototype illustrating advanced design concepts, system designers can experience first-hand the strengths and weaknesses of these design concepts. This experience can then be used as a basis for improving upon designs that are currently being explored in the commercial aviation community. (System developers from Northwest Airlines, Southwest Airlines, and American Airlines have already visited us in the past six months to get such first-hand experience.)
2. By creating a *polished* functional prototype, it is possible to run meaningful experiments to study the impact of design features on performance. Too many prototypes are developed that have shoddy interfaces which then hide the true effects of the underlying design concepts. We have therefore paid close attention to the craftsmanship of our system design at all levels of detail.
3. By conducting a large-scale empirical study which contrasts alternative designs, and by developing cognitive models to describe how these designs affect performance, we hope to provide insights that will change the questions that designers ask about their own designs.

Our goals, then, are to illustrate specific design features that may be of immediate value to the airlines when developing flight planning systems, and to provide guidance in improving the design process for developing aviation systems in general.

Methods

In order to conduct this research, we have developed the Flight Planning Testbed (FPT), a fully functional testbed environment for studying advanced design concepts for tools to aid in flight planning. Details about FPT are given in the paper attached in Appendix A. Certain features merit special emphasis:

1. As a testbed environment, FPT makes it possible to vary design features in order to conduct rigorous empirical studies. Furthermore, FPT supports such empirical studies by automatically logging all subject actions and the times of these actions as the subject uses the system;
2. To avoid potentially confounding conflicts between subjects' expectations about flight performance parameters and the behavior of FPT, the system has been designed to simulate the fuel consumption and speeds of a 757;
3. As a prototyping environment, FPT allows us to study advanced design concepts, rather that constraining us to study the rather crude flight planning systems currently in use at the various commercial airlines. To the best of our knowledge, FPT provides the most advanced design concepts of any flight planning system in the world at the present time;
4. As a prototyping environment, FPT also gives us control over the details of the system design, including its interface. This is important because a poorly designed interface could obfuscate the issues that we really want to study by interfering with planning behaviors. Consequently, we have very carefully crafted the design of FPT and gone through several stages of empirical testing to ensure that interface design features will not interfere with the more general issues we are studying.

Additional details on the design of FPT are given in Appendix A.

Experimental Design

We are studying performance using three different system designs. The first design requires pilots and dispatchers to explore alternative routes on their own using our graphical interface. The second includes access to the same graphical interface, but provides access to an additional tool in which the planners can specify constraints on a solution (maximum turbulence, maximum precipitation, destination) and ask the computer to find a path that minimizes fuel consumption while meeting these constraints. In this second design, the computer provides flight plan information and recommendations only when specifically requested by the planner. The third design under study is the same as the second, except that the computer automatically displays a recommendation.

Subjects. Our study includes 30 commercial airline pilots and 30 airline dispatchers. At present, we have volunteer subjects from 10 airlines, covering a wide range of flying and dispatching experience and different aircraft.

Procedure. A between-subjects design is being used in which 10 pilots and 10 dispatchers are being randomly assigned to each of the three system designs. Each subject is trained to proficiency using two training tasks and then tested on four carefully designed scenarios. The training requires approximately two hours. During the test scenarios, the subject is asked to think aloud as he develops alternative flight plans.

Following completion of the four test scenarios, each subject is then debriefed. In addition to collecting biographical data and subjective responses to the system, the subject is asked to evaluate the full range of possible solutions to each scenario, including those that he did not explore on his own.

All verbalizations and interactions with the computer are being videotaped. All actions and the times of those actions are recorded by the computer.

Data Analysis. In addition to studying the effects of system design on final answers, and in addition to studying the sequences of behaviors provided by the behavioral and verbal protocols, we have developed specific coding categories (identified prior to data collection). These coding categories are based on our predictions of subject behaviors when interacting with the different system designs. They include predictions regarding such things as potential fixations and the use of heuristics to focus attention on particular solutions, as well as predictions concerned with specific interface design features.

These data will be used to support the development of cognitive models of planning, to evaluate specific design features, and to contrast performance under the different system designs.

Preliminary Results

Thus far, we have collected data on 27 pilots and 22 dispatchers. We expect to complete data collection on 3 more pilots and 8 more dispatchers in January, 1992. Below we summarize some preliminary results for the 27 pilots (whom we are analyzing first).

Individual Differences

The data make it abundantly clear that there are significant differences in the preferences of different pilots for particular plans. On Scenario 4, for example,15 pilots preferred deviating north of the storm (an isolated supercell over Dallas), 9 preferred deviating south, and 3 wanted to stick to the original route but at a higher altitude (trying to fly over the storm). The data provided rich insights into the sources of these and similar individual differences on all four scenarios:

> "It's a little bit quicker and we aren't going to have any turbulence. We're going to get there a little sooner. The distance is less."

> "The winds are more favorable with the southerly route."

"It's a trade-off, you know, but I think staying out of the turbulence and giving the passengers a comfortable ride is better. Better to use a little fuel for that."

"Got some wind out of the south that might move some of that stuff out of the way."

"Shoot the dispatcher next time you see him for sending you right into the middle of the thing."

"This is what's forecast to happen, but the thing that's going to go through your head is: Where did this idea of the forecast come from and how reliable is it?"

"We should be through it before that hail crops up."

"It's 3 minutes longer to the south, but that stuff's moving to the north."

"We've got a lot more options if we go to the west of that storm activity."

"Given the usual traffic patterns, we're better off going south."

We expect the data to provide us with insights into the factors which should be considered in deciding whether a particular flight plan is a good one. The data also suggest that pilots differ in terms of their models of the world (regarding weather and traffic), the factors that they consider when evaluating a plan, and their priorities in evaluating these different factors.

These results have significant implications for training (in cockpit resource management sessions, for instance), particularly when we contrast the behaviors of pilots and dispatchers. They also have important implications for system design.

The Impact of Alternative System Designs

As discussed earlier, we are studying the effects of 3 alternative system designs on performance. In the first design (Sketch Only), the pilots and dispatchers must use our graphical interface to sketch alternative routes on their own. In the second design (Sketch plus Constraint-Setting), the subjects can sketch their own route and also set constraints on the desired solution and then ask the computer to find a solution. In the third design, the computer automatically displays a recommendation for an alternative plan (automatic suggestion). The subjects can then sketch other alternatives or change the constraints and then ask the computer to find another solution.

The data strongly suggest that the design of the system affects both the exploration and the evaluation of alternative plans. In Scenario 3, for instance, 4 of 10 pilots in the Automatic Suggestion condition selected what they themselves concluded in the debriefing was a very poor choice. (Scenario 3 was specifically designed so that the computer would initially suggest a poor plan.) Only 1 of 7 pilots in the Sketch Only version selected this bad choice and 0 of 9 in the Sketch plus Constraint-Setting condition selected it.

We are currently analyzing the behavioral and verbal protocols to better understand how the alternative system designs influence the subjects' cognitive processes.

More broadly speaking, then, it appears that our data will provide valuable insights into how alternative system designs influence the cognitive processes of the human planners and thus impact the final choice of a flight amendment.

Implications for Design

In addition to reporting the specific findings, we hope to generalize from our results to make statements about effective design processes. In particular, we expect to illustrate how general design guidelines are at best insufficient to ensure a good design, and at worst are misleading. We currently feel that such principles, if not supplemented with detailed, context-sensitive cognitive task analyses, are of marginal value. An example of one such analysis is provided below.

Principle 1: Avoid excessive automation of complex problem-solving tasks by keeping the person "in the loop."

If a computer could be designed that was a perfect problem-solver for the class of tasks of interest, and if this computer could be guaranteed to always be available and to always provide its answer in a timely fashion, this principle would be silly. The argument in favor of this principle, then, is that system designers, programmers, and hardware are all fallible. Regarding the fallibility of designers, the argument is that:

A. The designer may not identify all of the types of problems or situations that could arise. Consequently, the system she designs may be fallible for some unanticipated set of problems;

B. The designer may not correctly reason through how her computer will respond to each type of problem (because of time/resource constraints or because of the complexity of the problem-solving task). Again, the system she designs may therefore be fallible for some set of problems;

C. The designer may choose to develop an imperfect problem-solver because of time/resource constraints or because of the limitations of the available technology. In one case, for example, the designer may choose to develop a flight planning system that finds routes that minimize fuel consumption, but that ignores weather considerations. In another case, the designer may simply not know how to incorporate reasoning about uncertain weather forecasts into the computer's considerations, or how to deal with tradeoffs between safety and cost;

D. The designer may have developed the necessary knowledge base to correctly design the computerized problem-solver, but may slip (Norman, 1981) in applying this knowledge to specify the actual design.

Similar arguments apply to the programmers who implement the system (and indeed to any participants in various stages of the design process - including usability testing).

Counterargument to Principle 1. Unfortunately, system designers are not the only people who are fallible. The people they are trying to help, such as dispatchers and pilots, are also fallible. These people are also limited by the rate at which they can generate solutions.

Consequently, we are faced with a tradeoff. For most real, complex problems such as flight planning, we know that if a designer tried to fully automate the problem-solving task, the result would be unacceptable. Likewise, human planners "on their own" are likely to be fallible when performing a complex task like flight planning. This is true whether we define fallibility in terms of a failure to find the "best" solution in a timely fashion or in terms of a less stringent requirement to find a "satisfactory" solution in a timely fashion. It is also true whether we think of the "solution" as a static, one-time decision or a plan that is adapted over time as the situation unfolds (Suchman, 1984).

To make rational choices among alternative designs, then, principles like "avoid excessive automation of complex problem-solving tasks by keeping the person in the loop" are of very limited value. They point to a design decision that must be considered, but they don't really tell us the answer. The answer will in general be very context dependent and will require careful consideration of the relevant tasks and task environments, technologies, and the cognitive processes of both system operators and the system development team.

Consider, then, some of the questions that should be asked when designing a computer system to aid in problem-solving:

1. What different types of situations or problems can arise? How likely are they to arise?

   Note that this taxonomy of tasks must be sensitive to the characteristics of computer system's development team and process, the actual design of the computer system, and the human operator (if a person is involved in the overall system functioning). This demanding requirement arises because we are trying to identify situations where the designers, operators, and/or the computer system itself are likely to be fallible.

2. How likely is it that our task taxonomy is incomplete? How serious are the possible consequences of this incompleteness?
3. How will the computer perform in each of these different types of situations?
4. If a person is part of the overall system functioning, how will different people (in conjunction with the computer) perform in each of these different types of situations?
5. How likely is it that our predictions of performance are fallible? How serious are the consequences of such incorrect predictions?
6. Considering all of the above questions, and considering the cost of developing, operating and maintaining the proposed system, what is the expected utility (Raiffa, 1979) of this system design as compared to other alternatives?

In short, we need to go beyond vague principles like "Keep the person in the loop" and identify ways to answer the above questions that are sensitive to the specifics of a particular problem-solving task, to the design of a particular computer system, and to the characteristics of the people who will be interacting with this computer system.

Application of Principle 1 to Flight Planning. No one currently has the technology nor the resources to build a perfect computerized problem-solver for the task of enroute flight planning. Consequently, we have to consider the tradeoffs between different levels and types of computer automation or aiding in terms of fallibility, cost and the timeliness of getting solutions to problems. Our experiment, in which we are studying the alternative system designs, should provide data to help illustrate such tradeoffs.

Conclusion

Our report on this current experiment will provide data pertinent to several interesting questions:

1. What models are necessary to account for the planning behavior of various pilots and dispatchers? What strengths and weaknesses are indicated by these models?
2. How effectively can pilots and dispatchers interact with the information displays, graphical interface, and support functions provided by the FPT?
3. How do alternative system designs influence overall performance? How are the cognitive processes of pilots and dispatchers changed as a result of system design? What impact do these changes have on the quality of the plans developed?
4. What implications do these cognitive models have for designing effective cognitive tools to support planning?

Thus, we feel we have identified a number of important research issues and design concepts relevant to the development of cooperative problem-solving systems in general, and specifically to the design of flight planning tools. As one dispatcher (who had seen the report in Appendix A as well as FPT) commented:

> "I have just finished reading your technical report on 'Design Concepts for the Development of Cooperative Problem-Solving Systems.' It is truly great stuff! Your observations on the flight planning program and process and my 14 years of airline dispatch experience are just about in 100 percent agreement."

FPT provides a powerful environment for studying these advanced design concepts. We are therefore using it to complete an unusually rigorous empirical study of the performances of pilots and dispatchers as they interact with alternative system designs. The full results of this study should be available in March, 1992.

Appendix

Design Concepts for the Development of Cooperative Problem-Solving Systems

Chuck Layton
Philip J. Smith
Cognitive Systems Engineering Laboratory
Department of Industrial and Systems Engineering
The Ohio State University

Elaine McCoy
Department of Aviation
San Jose State University

Tom Bihari
Adaptive Machine Technologies, Inc.

Introduction

There are many problem-solving tasks that are too complex to fully automate given the current state of technology. Nevertheless, significant improvements in overall system performance could result from the introduction of well-designed computer aids.

We have been studying the design of cognitive aids for one such problem-solving task, enroute flight path planning for commercial airlines. Our goal has been two-fold. First, we have been developing specific system designs to help with this important practical problem. Second, we have been using this context to explore general design concepts to guide in the development of cooperative problem-solving systems. These design concepts are described below, along with illustrations of their application.

The Application Area

Before take-off, a complete flight plan is developed describing the route, altitudes and speeds that a commercial airliner is expected to follow in flying from its origin to its destination. This initial flight plan is rarely followed exactly as specified prior to take-off, however. Minor amendments to the plan are common; major changes are not at all unusual.

Such replanning of the flight while enroute arises because of the dynamic, unpredictable nature of the "world" that must be dealt with. Weather patterns do not always develop as predicted, resulting in unexpected areas of turbulence, less favorable tailwinds, or storms that must be avoided. Air traffic congestion may delay take-off or restrict the plane to lower than planned altitudes while enroute. Airport or runway closures can cause major disruptions. Mechanical failures, medical emergencies or other critical problems may force the plane to divert to a nearby airport.

A Problem Space Description

Enroute flight planning can be represented as search through a problem space (Laird, Newell and Rosenbloom, 1987). When some problem arises, as described above, the flight crew, Dispatch and Air Traffic Control must develop a revision of the flight plan. To generate this revised plan, a variety of alternative solution paths may be considered.

A state description for one of the possible problem space descriptions consists of:

1. The plane's current location (a point along its route and an altitude) and airspeed;
2. The plane's currently approved flight plan;
3. Static and dynamic characteristics of the plane such as its weight (which changes as fuel is consumed), its maximum altitude capabilities (which change as a function of the plane's weight and airspeed), its fuel consumption characteristics, etc. Characteristics that are normally considered static may in some cases change because of a problem like engine failure;
4. Actual and forecast weather along the plane's current path and any possible alternative paths. The state description needs to include measures of uncertainty about weather forecasts, as well as the best "guess";
5. Information on passenger connections and flight crew availabilities;

6. Static and dynamic characteristics of airports that could be used for landing (runway lengths, visibility, air traffic congestion, etc.);
7. Similar information for any other planes whose paths could interact with possible alternative paths for the plane that is the focus of the replanning activities.

(This is a simplified summary of a state description. Each of these components are actually composed of many additional elements.)

Major operators include:

1. changing altitude;
2. changing airspeed;
3. changing the route;
4. changing the destination (a special but important case of changing the route).

Each of these operators can be applied to either the plane that is the primary focus, or to some other plane that its plan interacts with. Furthermore, the first three operators can be applied to different segments of the flight. The plane may fly at 33,000 feet from Milwaukee to Chicago, but at 25,000 feet from Chicago to Toledo.

There are also a number of constraints. Planes must maintain a certain separation distance (to comply with FAA regulations). Planes fly along "highways in the sky". (They fly from waypoint to waypoint to get to some destination, instead of flying straight to that point. They are also constrained to fly at certain altitudes. Over the continental U.S., for instance, 33,000 feet is an "eastbound only" altitude.) There are also certain physical limitations. The plane can't fly if it is out of fuel and it can't land at an airport where the runways are too short. Some of these constraints are actually "soft". If, for instance, there is no traffic, Air Traffic Control (ATC) may allow the plane to fly west at an "eastbound only" altitude. Similarly, ATC may approve a vector that deviates from the waypoint to waypoint "highways" in order to avoid a storm or save on fuel.

Description of the state spaces, operators and constraints are difficult because there are so many possibilities to consider. Definition of the evaluation function for selecting among operators is even more challenging, however. It is clear that multiple competing and complementary goals are considered (Wilensky, 1983) in evaluating preferences among alternative operators (or operator sequences). Safety, fuel consumption, time and passenger comfort are all important considerations. It is not so clear, though, exactly how human planners currently deal with tradeoffs among these goals.

In short, the full problem space for enroute flight planning is very large and complex. Multiple goals must be considered in a highly stochastic environment where multiple plans must be coordinated.

Cooperative Problem-Solving as a Conceptual Approach

Our conclusion, based on this initial problem space analysis, was that complete automation is not likely to be an acceptable approach for improving such planning performance. Neither knowledge-based systems techniques nor optimization methods are sufficiently trustworthy to fully replace human judgment on such a planning task. Concerns over inappropriate model formulation, incompleteness of the knowledge-base, brittleness in dealing with novel situations, difficulties in trading off among competing goals and inadequacies when making decisions in uncertain environments all introduce significant objections to complete automation as a solution.

One approach to alleviating such concerns is to try to build better computerized problem-solvers. Current work on "deep reasoning" systems and qualitative modeling falls into this tradition.

An alternative (but complementary) approach is to focus on shared problem-solving. This, in fact, was one of the major motivations behind much of the early work on expert systems. In response to failures and lack of acceptance for problem-solving systems based on optimization techniques, the artificial intelligence community suggested the design of systems that solved problems "like people do". Michie (1986), for example, states:

> "The suggestion is that reliance on the escalating power of brute force may be heading towards danger. However effective and reliable such systems may be in normal conditions, use of brute force may not be worth the price paid during the rare episodes when a computer-controlled power station or military installation or air-traffic control system malfunctions. On these occasions, a new

factor becomes paramount. The human operator or supervisor needs to follow what the computing system thinks it is doing."

Early work on expert systems, as a reaction to optimization approaches, set out to increase the cognitive compatibility of computer problem-solvers and their users by attempting to mimic human cognitive processes. This is only one of many concepts, however, that are useful in guiding in the design of more effective cooperative problem-solving systems.

Below, we describe additional design concepts that have guided our work in developing a cooperative planning system. Equally important, we illustrate the importance of understanding not only how people correctly solve particular kinds of problems (Smith, Smith, Svirbely, Galdes, Fraser, Rudmann, Thomas, Miller, Blazina and Kennedy, in press), but also the nature and causes of errors that people make in solving these problems (Fraser, Smith and Smith, 1989), and the ways in which alternative system designs influence and enhance shared problem-solving.

Initial Studies

In order to better understand human performance on flight planning tasks, we began by:

1. Interviewing pilots, air traffic controllers and dispatchers. (Dispatchers work for individual airlines and are responsible for

developing the original plans for flights and for helping the flight crew to generate amendments while enroute.);

2. Conducting a survey of 136 pilots to identify situations where they had experienced problems with enroute flight planning (Smith, McCoy and Layton, 1989);
3. Running studies in a flight simulator to observe actual flight planning activities (Galdes and Smith, 1990).

These studies made it clear that enroute flight planning activities are currently distributed among the flight crew, ATC and Dispatch. They also made it apparent that, at present, flight crews play a major role in detecting situations that require replanning, in generating possible flight amendments and in evaluating the alternative plans. ATC may help generate details of a plan (when the flight crew makes a request like: Can you vector us north of this storm?) ATC also places constraints on the acceptability of alternative plans. If the presence of other air traffic makes a plan unworkable, ATC is responsible for noting this. Depending on the circumstances, Dispatch may be uninvolved, or may do most of the plan generation (finding a suitable alternate destination, for instance). Examples of behaviors observed in our simulation study are given below.

<u>Example 1</u>

Fifteen minutes after takeoff, the pilot requested clearance to climb from FL 250 to FL 290. ATC denied this request because of other traffic. In response to this event, the flight crew did the following:

1. Asked ATC how long they would be at FL 250.
2. Noted that they "ought to call Dispatch and tell them we're at a

different altitude", but chose not to call Dispatch yet.

3. Asked themselves: "What do you think our difference in burn would be at 250?"
4. Determined the differences in fuel burn and time (actual vs. planned) at the next waypoint : "47.7--we're 200 pounds under."
5. Checked the wind speeds and directions: "Have the winds changed at all? We're coming up on Mustang. Mustang has winds at 290 of 44 knots."
6. Predicted the extra fuel burn resulting from staying at FL 250 until Battle Mountain (the point at which ATC had indicated they could probably climb): "I guess we know we're going to burn some more fuel staying down here, but probably as much as 500 pounds maybe."
7. Further evaluated the implications of staying at FL 250: "Twenty-five minutes down here. That'll let us get to 33 a little ahead of time because we'll have burned off fuel just a little ahead of time. Yeah. Possible. I don't know."
8. Planned their next change in path: "Battle Mountain. That's when I'm hoping to get 29,000."
9. Evaluated this plan by checking the winds at Battle Mountain.

As this example illustrates, the flight crew was extremely active in considering alternative flight paths. They collected a variety of data to determine the implications of the unplanned deviation from their route, and to decide what they should do next. Some of this data involved comparing actual performance (e.g., fuel burn) with that expected under the original plan. Other data required making predictions about future performance if the current altitude was maintained.

<u>Example 2</u>

In the first example, ATC instructions made it necessary for the pilots to consider the implications of a different route. In this second example which occurred 54 minutes into the flight simulation, one crew detected data that caused them to consider a different route for other reasons:

1. Looking at a radar display, the co-pilot noted: "We could have some activity on the way to Detroit, too. I think we're going to want to go north of that. North or south. It looks like north would be better."

2. The crew then proceeded to develop such a plan: "It seems like maybe we could reroute our flight up above there [North] rather than wait 'til we get up here... What kinds of VORs are we looking at then? Should we maybe go to Aberdeen flying up north and possibly Redwood Falls?"

3. The pilot then requested such a change: "We have a routing request we'd like to have you pass on to our dispatcher. We'd like to fly Jet 32 to Aberdeen, then Jet 70 to Badger. We'd like to remain at FL 250 for the time being."

This example again illustrates the fact that the flight crew currently plays an active role in detecting the need to consider an alternative plan and in generating the alternative plan.

Example 3

Two hours and sixteen minutes into the flight, the same crew as in Example 2 began to consider the thunderstorm again:

> "That looks kinda nasty. We tried to tell them a long time ago we wanted to go north of that. I'm not wild about going between those things. There's not 20 miles between them. I vote total deviation. Ask 'em for a vector around the north side of the weather. How far are we going to have to go? 100 miles? If we start down, we won't have to go as far out of our way. Just tell 'em we want to vector north of the weather and let them [ATC] do it. We don't have enough information to be that specific. There's no way we're going to fly into that... Holy shit! There's stuff behind it, too. Holy Mother!"

This example provides a nice illustration of the role of the crew in detecting a problem and considering alternatives. It also points out the importance of coordination between the crew, ATC and Dispatch. In particular, the crew noted, "Taking our deviation a lot further back would have made a whole lot more sense."

Example 4

Two hours and forty-eight minutes into the flight, one crew began to worry about their destination:

> "I have a bad feeling about Detroit. Should have been starting to

clear... The minimum there - we need a half mile... What did they show for the fuel there? 18.6 - One thousand pounds less than original... I recommend, gentlemen, if Detroit doesn't look good we go direct to Cleveland and we go to the 100 Bomb Group for dinner, to the restaurant right next to the airport... Chicago's pretty good. Milwaukee's not bad. Our landing fuel just gets lower and lower."

Based on such data, and on the results of our interviews and surveys, we completed a cognitive task analysis (Galdes and Smith, 1990). This identified pertinent goals, data and problem-solving activities, as well as providing insight into the roles of the various players. It also identified problems arising in existing planning environments, ranging from failures to detect problems with the current flight plan in a timely manner, to inadequate generation of alternative solutions (thus missing a good alternative), to fixation on a potentially dangerous solution.

We then used this analysis as the basis for designing FLIGHT PLANNER, a prototype system to aid in enroute flight planning. Below we:

1. Describe the prototyping environment built to support system development and testing;
2. Present our initial implementation of FLIGHT PLANNER;
3. Discuss general design concepts that guided us in the development of this cooperative problem-solving system.

As part of these discussions, we also point out important insights that arose from our cognitive task analysis.

Development of a Prototyping Tool

Our research plan calls for a two-stage approach to testing design concepts. The first stage involves the use of a part-task simulation in order to develop design concepts and complete an initial evaluation. Those concepts that prove most promising based on this initial evaluation will then be used in the second stage of testing. This second stage will involve evaluation in the NASA Ames Advanced Concepts Simulator.

In order to run experiments using a part-task simulation, we had to design a suitable development environment. We consequently built a prototyping tool that can support the development and testing of a variety of design concepts.

This prototyping shell, designed to run on a Mac II, provides a general environment for developing application software, but does not inhibit programmers from modifying the environment if necessary. Written in Lightspeed C, the system can control displays on up to six color monitors.

This prototyping tool supports the creation and use of multiple window displays on each screen and the use of both mouse and keyboard inputs. The tool also provides both real-time and simulation-time clocks to control the timing of events and to record response times. The system records the time and nature of all actions made by a subject, and can replay the entirety of a subject's actions at a later time.

Development of FLIGHT PLANNER

Using this prototyping tool, we have been exploring a number of design concepts in a system called FLIGHT PLANNER. (See the first photo). This prototype system provides aids for enroute flight planning. It has several important features which are described below.

Map Display

FLIGHT PLANNER is capable of generating an accurate map display for any portion of the world. To accomplish this, we have ported to the Mac II a program (and associated database) that was developed using data from the U.S. Geological Survey. This program can produce accurate displays of any portion of the world, using any one of several available map projections.

FLIGHT PLANNER also allows for easy, rapid display of weather information on this map display. By simply pressing buttons with a mouse, the pilot can select a variety of weather overlays (radar weather, jet streams, fronts, etc.) to display on the map. (See the second photo). In this manner, the planners (pilots or dispatchers) can personalize the weather display to meet their current needs. Furthermore, by double-clicking with the mouse on any portion of the map display, the planner can zoom in on the region, seeing a close-up display.

In order to facilitate viewing trend information, the planner can also view weather sequences over time on the map display. This is accomplished by moving the plane along its route on the map. The plane is moved using a scroll bar controlled by the mouse.

The map display can also show weather information at different altitudes. (The National Center for Atmospheric Research has indicated that such data will be available nationally within the next five years.)

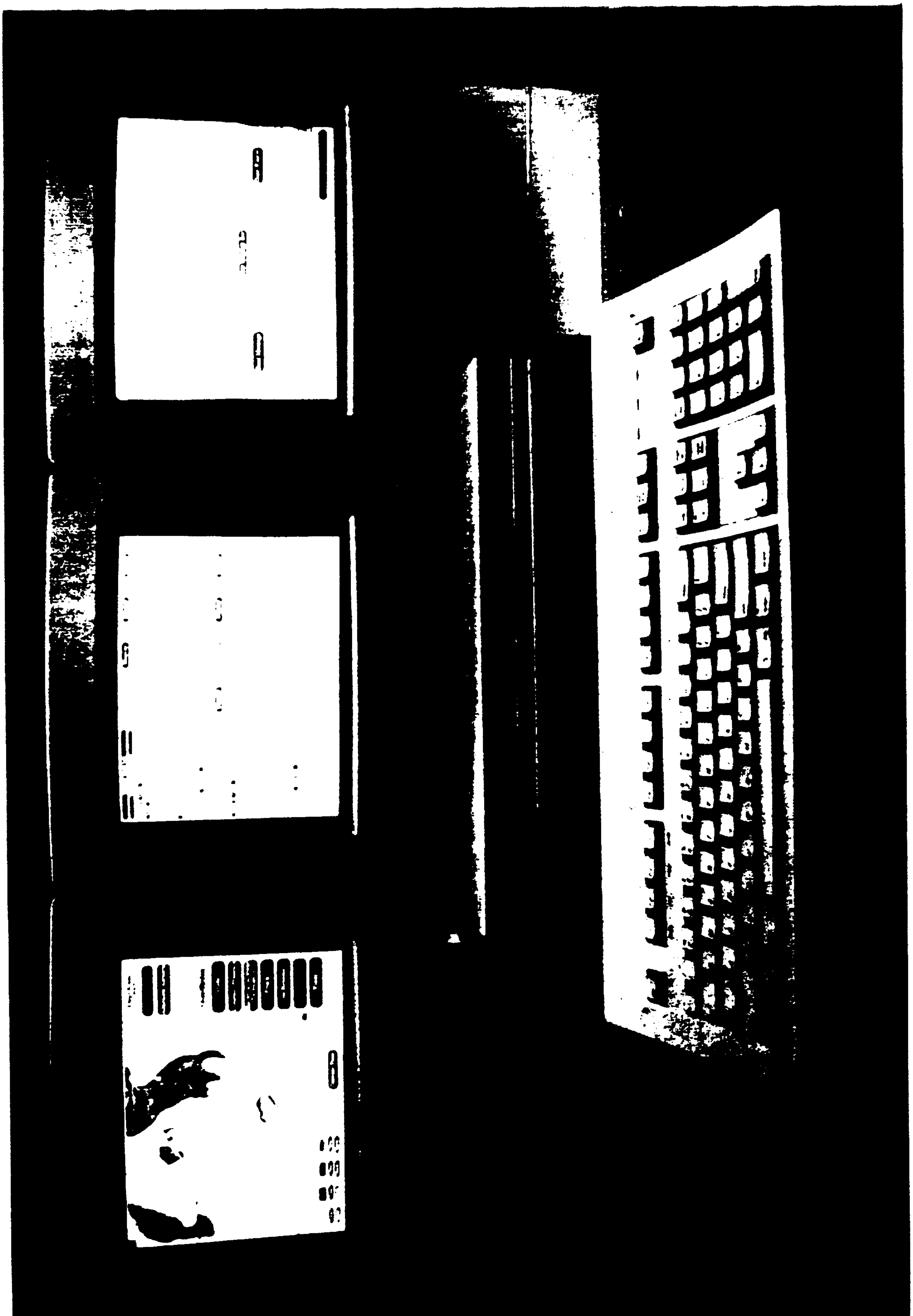
In addition to presenting weather information, the map display can show up to four alternative routes for the plane. It also displays the location of the plane on the active route. Both the plane's location and the weather displays are updated over time during the simulation.

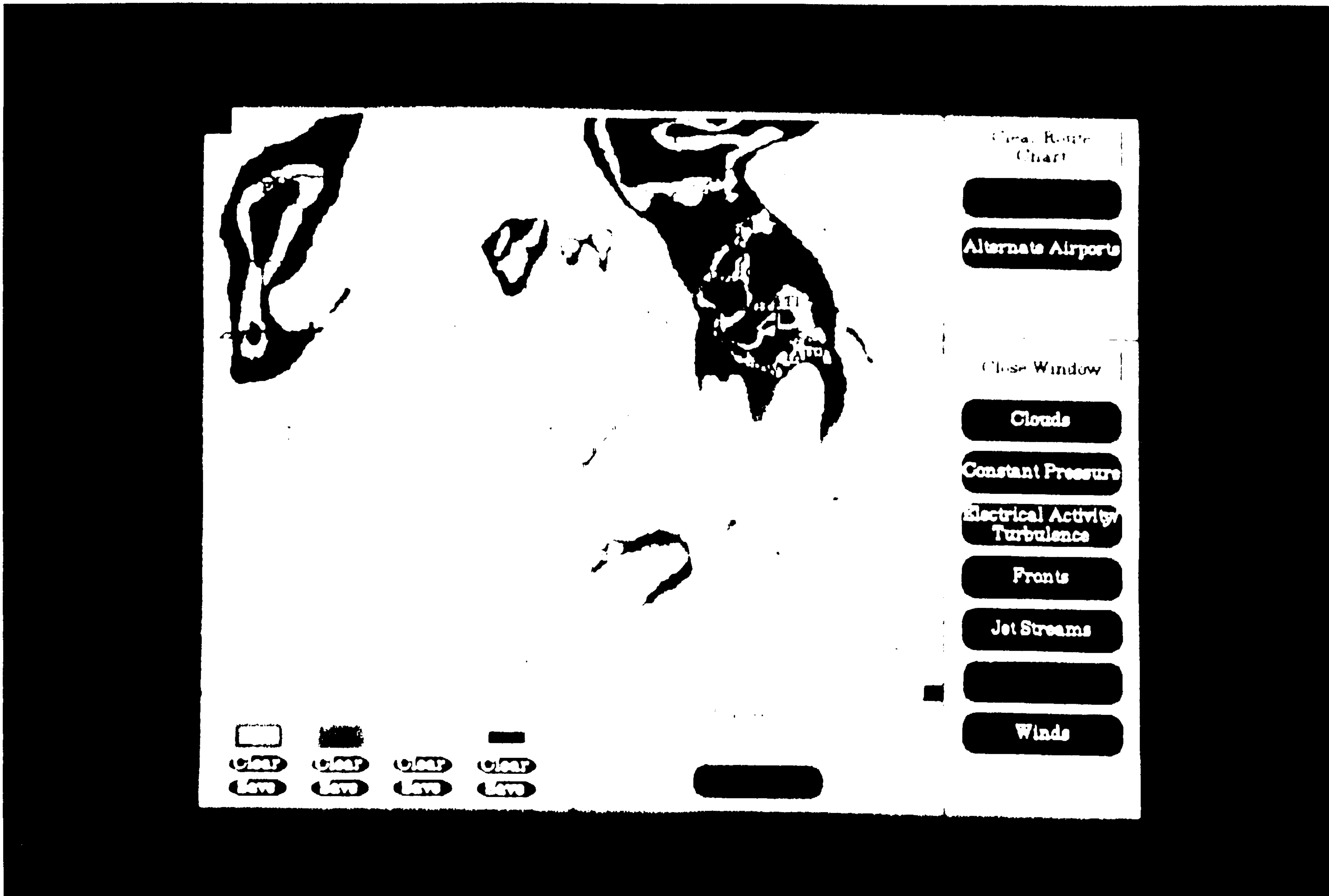
Routes can be created or changed on the map display in two ways. One way is by direct manipulation of routes on the map itself using the mouse. With the mouse, the planner can bend routes to deviate around some area. The planner can also create new legs off an existing path. Finally, the planner can create a totally new route.

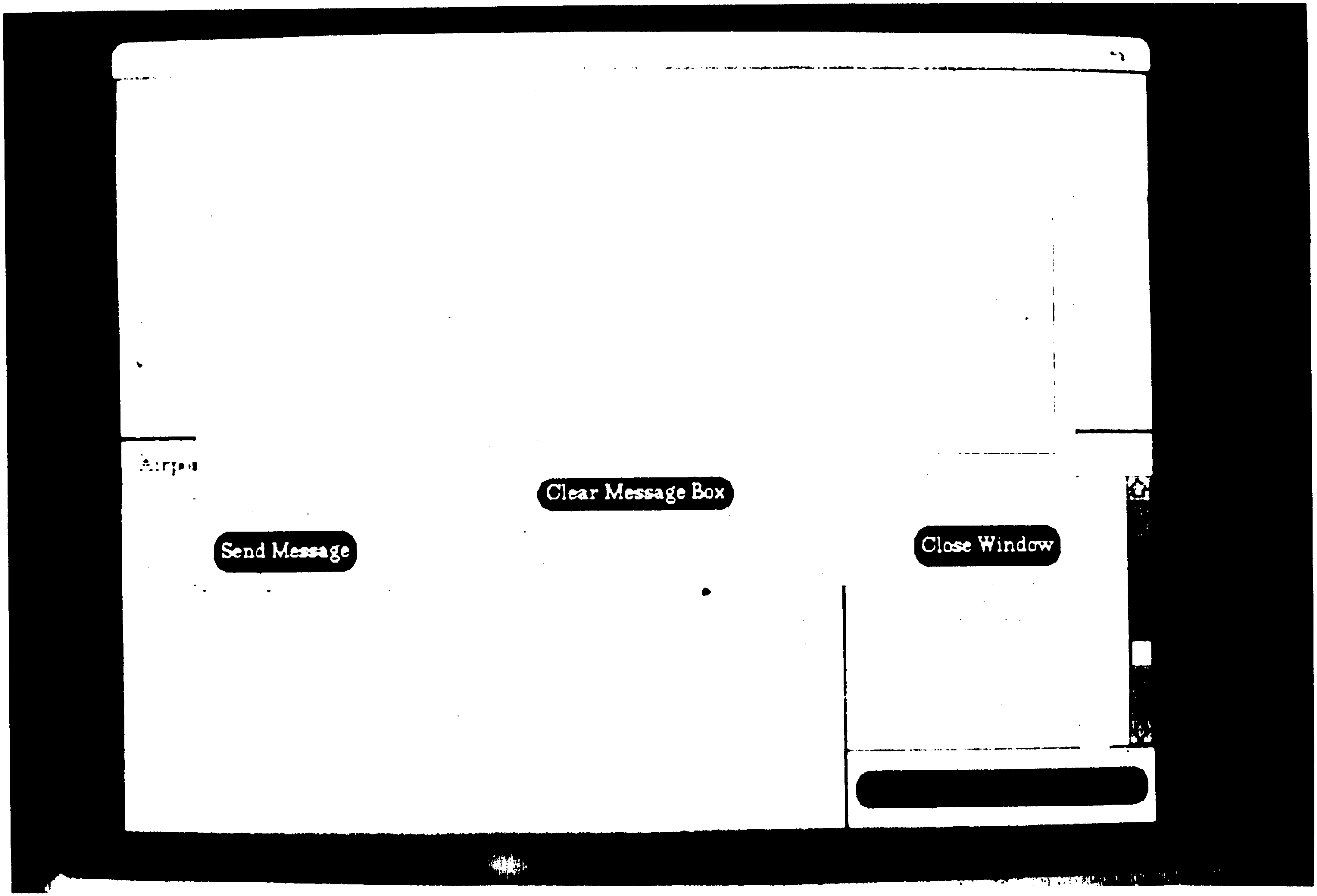
A second way to create or change routes is described in the section on the Route Information Display. In that window, changes to routes can be made using the keyboard.

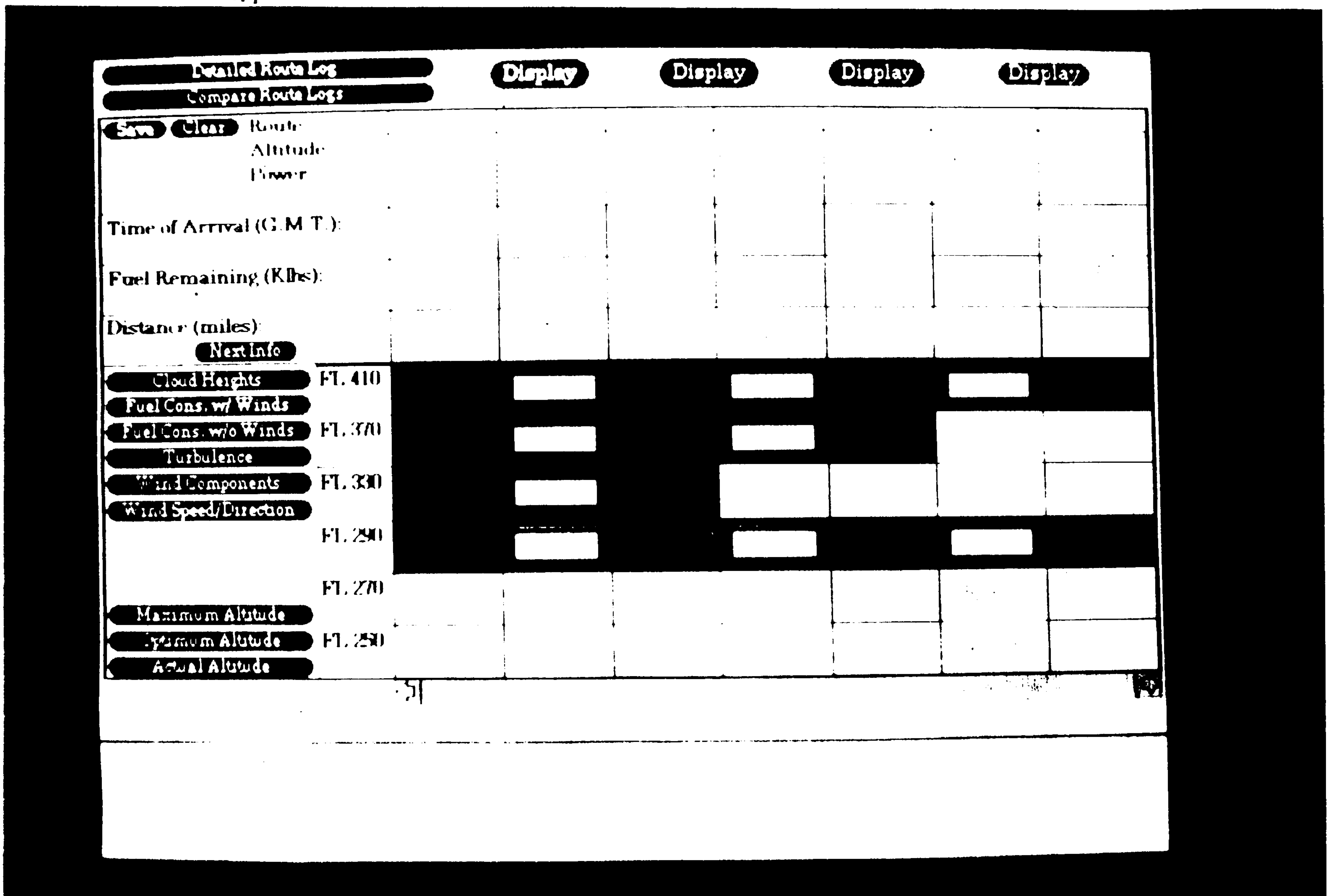
Information Alert Window

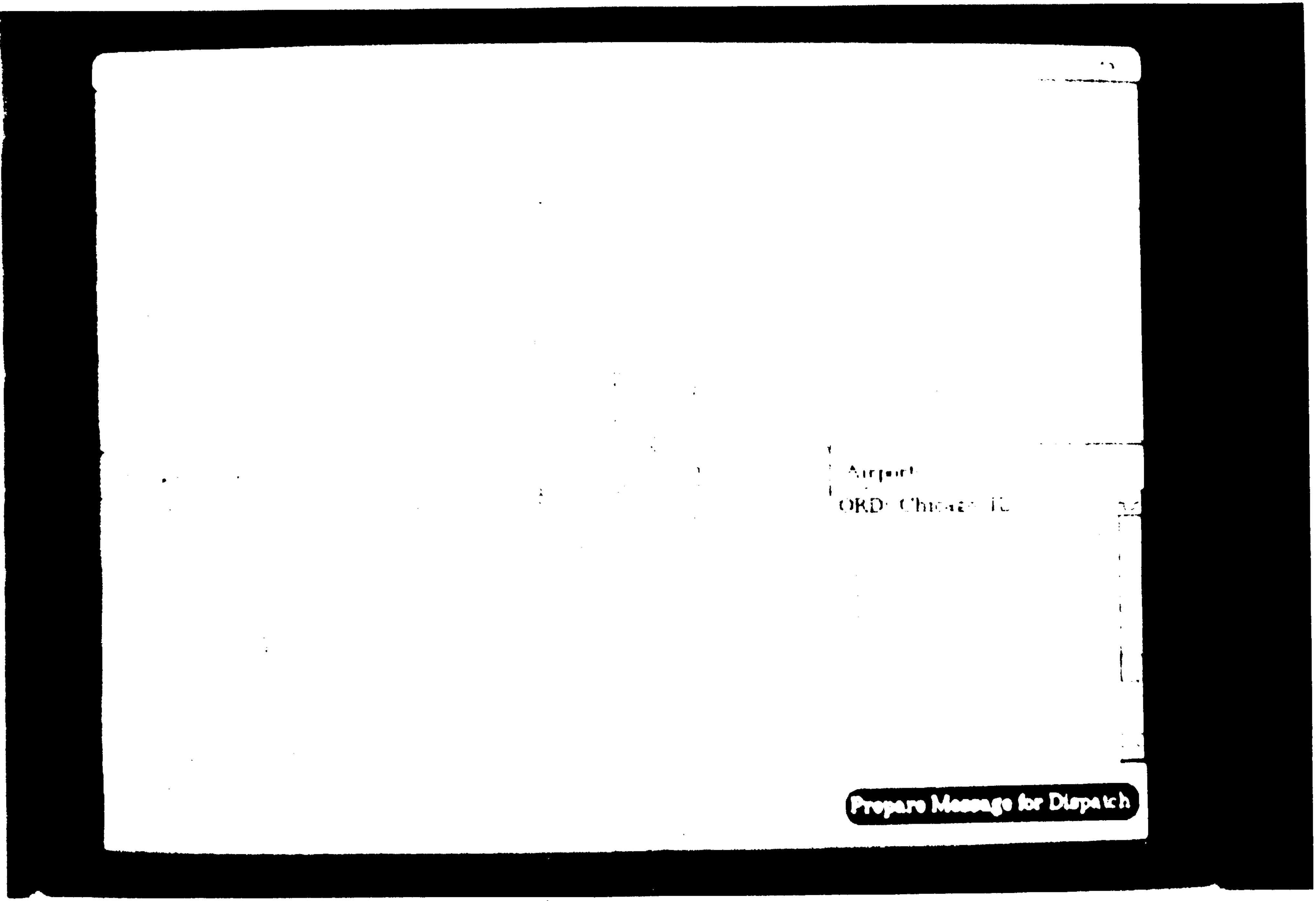
FLIGHT PLANNER also includes a window that can display important alerts to the planner at appropriate times during the simulation.

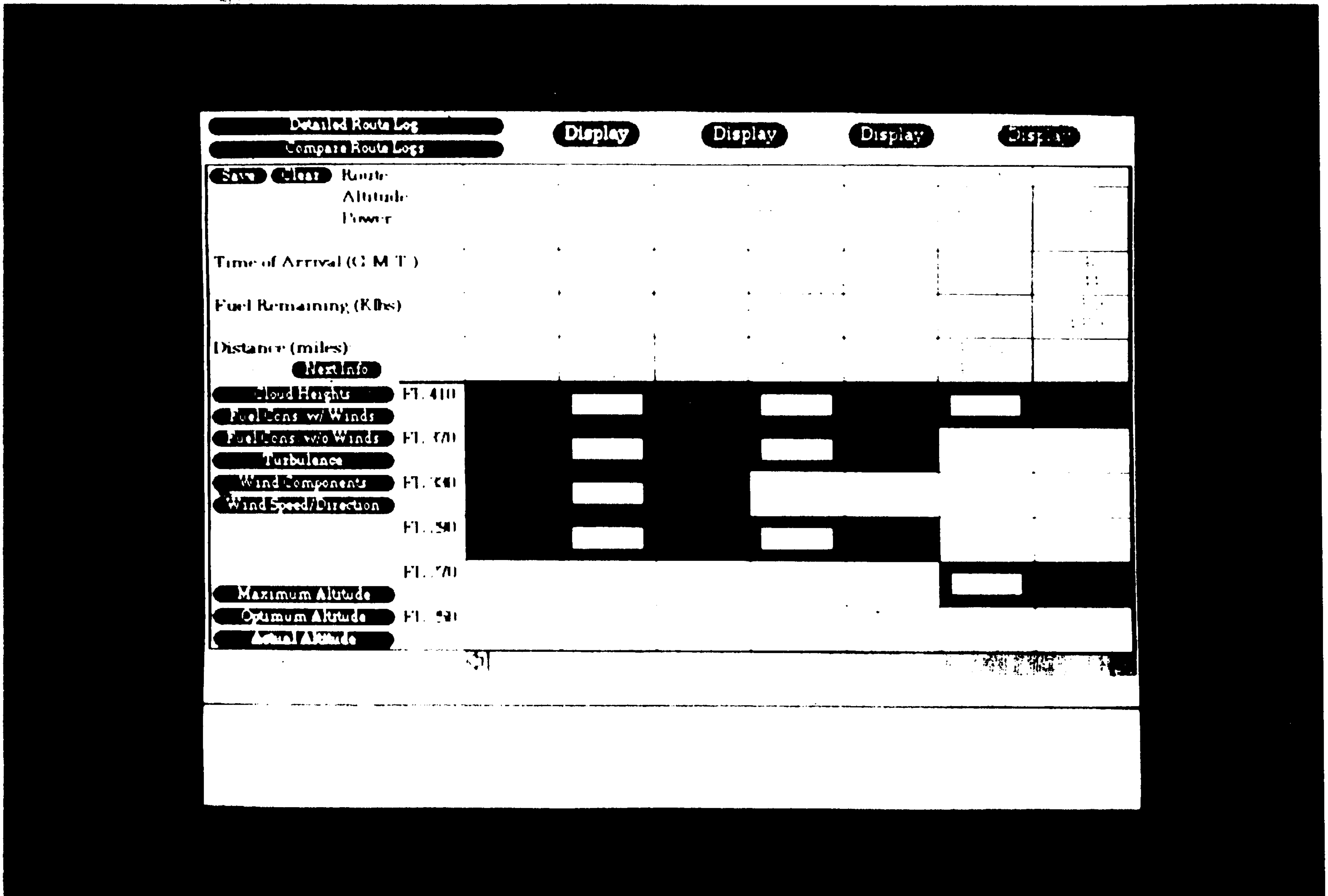



Close Window
Alternate Airports
Clouds
Constant Pressure
Electrical Activity/ Turbulence
Fronts
Jet Streams
Winds
Clear
Clear
Clear
Clear


Clear Message Box
Send Message
Close Window


Detailed Route Log
Compare Route Logs
Display
Display
Display
Display
Save
Clear
Route
Altitude
Power
Time of Arrival (G.M.T.):
Fuel Remaining (Klbs):
Distance (miles):
Next Info
Cloud Heights
Fuel Cons. w/ Winds
Fuel Cons. w/o Winds
Turbulence
Wind Components
Wind Speed/Direction
Maximum Altitude
Actual Altitude
FL 410
FL 370
FL 330
FL 290
FL 270
FL 250


Airport
ORD: Chicago IL
Prepare Message for Dispatch



Detailed Route Log
Compare Route Logs
Display
Display
Display
Save
Clear
Route
Altitude
Power
Time of Arrival (G.M.T.)
Fuel Remaining (Klbs)
Distance (miles)
Next Info
Cloud Heights
Fuel Cons. w/ Winds
Fuel Cons. w/o Winds
Turbulence
Wind Components
Wind Speed/Direction
Maximum Altitude
Optimum Altitude
Actual Altitude
FL 410

Communications Window

The system has another window that provides a text editing environment for preparing and sending written messages to other parties involved in planning activities. (See the third photo). Routes drawn by a pilot on the Map Display, for instance, can be transmitted to Dispatch along with text.

Airport Information Window

This window displays both static information (number of runways, etc.) and changing information (weather, NOTAMS, etc.) about specific airports. The planner can request such information by typing in the airport's identifier or by scrolling through an alphabetical list and selecting the airport with the mouse. (See the fourth photo).

Route Information Display

The Map Display provides a graphic presentation of weather data. There are other types of information, however, that are better displayed in a text format. We have developed a spreadsheet concept to present such information.

The fifth and sixth photos show a spreadsheet display available in FLIGHT PLANNER. Several important features are illustrated. First, the layout of data in the form of a spreadsheet seems well suited to this application.

The horizontal sequence of information on the spreadsheet corresponds to the horizontal sequence of waypoints and jet routes along the flight path. Information specific to particular waypoints and jet routes is displayed under the column with the corresponding waypoint or jet route label.

Second, the spreadsheet allows the planner to immediately view the implications of a change in the flight plan. The planner can make changes in the plane's route on the spreadsheet by simply adding or deleting the appropriate waypoints. These changes in the route are immediately drawn on the Map Display. (Alternatively, the pilot can change the route by direct manipulation of the path shown on the Map Display. These changes are propagated to the spreadsheet.) The pilot can also make changes in the planned altitudes and airspeeds on the spreadsheet.

When a change is made in the flight plan, the system will appropriately change the other information displayed (such as arrival time and fuel consumption). The spreadsheet allows the planner to view a variety of such information, such as wind components and distances between waypoints, as well as fuel consumption and arrival time information. Summary information is provided at the bottom of the screen for all routes that have been created, thus facilitating comparisons among alternative routes.

The bottom half of the spreadsheet allows the planner to easily compare different information about altitudes along a route. The planner can display information such as turbulence, fuel consumption and wind components at these different altitudes. To facilitate such comparisons, the planner can display the current altitude profile, optimal altitude profile and maximum altitudes. These kinds of information are displayed graphically within the spreadsheet itself.

Intelligent Aids

There are four areas where the computer can use knowledge to make intelligent inferences and suggestions:

1. Determining a "good" route (sequence of waypoints), "good" altitudes and "good" airspeeds;
2. Inferring the intentions of the human planner in order to facilitate communication;
3. Alerting the planner when some important new data is available or when significant problems exist with a plan that he or she has proposed;
4. Helping the pilot to find a good alternative destination if the need arises.

These capabilities and associated issues are discussed in the context of the design concepts presented below.

Design Concepts

In studying the design of aids for enroute flight planning, we have encountered a number of relevant design concepts that apply. These are discussed below. The value of such a list of concepts (and examples of their applications) is their ability to stimulate the thinking of system designers. The designer must still consider his or her particular context in order to assess the applicability of a particular design concept, and to generate ideas on how to apply it to the specific problem area. By considering such a list of concepts, however, the designer of some new system may come up with solutions that might otherwise be overlooked.

Concept 1. Use data abstractions to help planners deal effectively with large quantities of data.

In the near future, the amount of information that could be provided to the people responsible for enroute flight planning could be greatly increased. Data about passenger connections, flight crew schedules and air traffic congestion is already available for use. In addition, the technology exists to provide detailed, frequently updated weather information. Every plane in the sky is a potential weather sensor transmitting data about turbulence, winds, etc. to ground stations. (United and Northwest Airlines are already experimenting with this.) In addition, wind profilers, NexRad, ACARS and automated weather stations will be available to provide further detailed weather data.

Three questions arise:

1. What data should actually be provided to planners?
2. How should this data be displayed and utilized?
3. Who should have access to what data (ATC or Dispatch or the flight crew)?

In this section we deal with one answer to the second question.

Consider a system where an international turbulence map is available and updated regularly. The quantity of data to consider is huge.

Clearly, the planner for a particular flight can begin by focusing attention on the airspace along that flight's route. With up to 20 flight segments for longer flights, however, the number of relevant pieces of data is still very large.

We need some way to help the planner focus attention on potential problem areas, and on likely solutions. Our current design illustrates one solution, using a data abstraction.

Consider the detailed spreadsheet display. The spreadsheet can display turbulence reports for each of several altitudes along the route. (See the sixth photo.) It also displays (as a colored line) the planned and optimal altitude profiles. (The planned altitudes are shown in the same color as the route; the optimal altitudes are shown in green.)

It would be impossible to display detailed turbulence data within such a compact display. Indeed, the pilots we have tested with our system indicate that, for just one individual flight segment, there could be considerable variation in turbulence levels at different points. Currently, such data is provided only in a detailed text format for pre-flight planning (e.g., "there is light turbulence along Jet Route 793 fifty miles east of

CMH").

We could simply create a listing of all the turbulence information for all of the points along the route for all of the nearby altitudes. Instead, we are using our spreadsheet display to present an abstraction of this turbulence information. The label (light, moderate, etc.) in the spreadsheet cell indicates the maximum turbulence level along that segment at that altitude (see the sixth photo).

Imagine a planner who wants to ask:

> Am I likely to encounter significant turbulence in the next segment of my flight?

This planner can simply scan along the altitude profile as displayed in the spreadsheet and see whether any of the flight segments show significant turbulence. If, for instance, one segment indicates moderate turbulence, he/she can click on that cell, opening a window which describes in detail the nature and extent of the turbulence along that segment.

Imagine this same planner asking:

> Can I avoid this moderate turbulence by changing altitude?

He/she can simply scan the spreadsheet cells, looking for an altitude corresponding to that flight segment that has less turbulence indicated.

Thus, FLIGHT PLANNER allows us to study the effectiveness of such data abstractions in helping the planner to detect potential problems in a timely manner, and to generate potential solutions. (An analogous form of data abstraction applies to the map display, where the planner can zoom in on a region and get more detailed information about weather and airport locations.)

This concept is particularly important in designing cooperative systems. The goal is to allow the computer and the human planner to *both* be actively involved in detecting the need to replan, and in generating and evaluating alternative plans. In order to critique the computer's suggestions and to generate alternatives of his/her own, the human planner needs access to the pertinent data in a usable form. It is not sufficient to simply provide the human planner with an explanation justifying the computer's recommendations. The assumption behind the design of a cooperative system is that there will be cases where the human planner will be capable of generating a better plan than the computer. Data abstractions offer one method for assisting the human planner in accessing the data necessary to accomplish this.

Concept 2. Allow direct manipulation of graphic displays to enhance exploration.

Our preliminary tests indicate that pilots are very enthusiastic about the ability to graphically create and manipulate routes. The ability to make the changes directly on the map display makes it much easier to explore alternate routes to avoid bad weather.

Using our map display, the planner can also move the plane along the route and watch the (forecast) weather change. This helps the planner to assess trends in the weather and their potential impact on the flight. It also helps the planner to answer questions such as:

> Am I likely to encounter bad weather at my destination?

If the answer to this question is affirmative, the planner may want to request extra fuel (if this potential problem has been noted before takeoff) or identify suitable alternate airports.

In the spreadsheet display, the planner can also manipulate the altitude profile graphically. He/she can simply drag the altitude profile up or down in order to explore alternative altitudes, rather than having to type in these alternative altitudes.

Like Concept 1, this design concept recognizes the importance of supporting the human planner in developing and evaluating alternative plans. Such uses of direct manipulation (Norman and Draper, 1986) make it easier to accomplish this goal by allowing the planner to explore alternatives by manipulating routes and altitudes on the data displays themselves.

Concept 3. Support planning and plan evaluation at many levels of detail.

Sacerdoti (1974) discusses the use of abstraction hierarchies to improve the efficiency of planning systems. Based on an analogy to this idea, we have developed a system where the human planner can develop plans at several levels of detail.

Flight planning is well characterized in terms of such an abstraction hierarchy. Imagine, for instance, a pilot flying from San Francisco to Detroit who learns of a line of thunderstorms crossing his flight path over the Plains States. His primary decision is whether to deviate north or south of this storm. In order to evaluate this choice, however, it is necessary to specify additional details. Waypoints, altitudes and airspeeds must also be specified.

In order to support this goal:

1. The pilot first sketches out a general solution (such as a northern deviation around the storm). This sketch is drawn on the Map Display;

2. By default, the computer fills in the lower level details, finding waypoints that approximate the pilot's sketch, finding an "optimal" altitude profile for this path and finding suitable power settings;

3. The pilot then evaluates the details of this solution by looking at the spreadsheet displaying route information such as expected

> arrival time and fuel consumption. If he chooses to, he can alter the computer's recommendations for the lower level details and compare his choices with the computer's. (He may, for instance, note that the computer's recommended altitude profile flies the plane through areas with unacceptable turbulence and therefore select a different altitude.)

Consider another situation where a pilot encounters turbulence. He/she wants to decide whether to go higher or lower. Using the spreadsheet display, he/she can directly generate and evaluate alternative altitudes.

Thus, we have designed a system where:

1. Displays exist corresponding to different levels of detail in the planning hierarchy;
2. The planner can view and make changes on any of these displays. The planner can change waypoints on the map display and altitudes or airspeeds on the spreadsheet. He/she can therefore make changes at any level of detail desired. He/she can also look at the data needed to evaluate decisions at that level of detail;
3. The computer, by default, handles lower levels of details. The planner can, however, compare the computer's recommendations with his/her own ideas and make changes as desired at any level of detail.

Thus, using this architecture, the planner can easily explore "what if" questions at any level of detail desired.

Note also that, for this part of the system, important issues begin to arise regarding the nature of the computer's planning processes. The planner may initially choose to rely on the computer's solutions at lower levels of detail while deciding whether to select a route north or south of the storm (as described above). At some point, however, the planner must decide whether to accept these lower level details as suggested by the computer, or to modify them. This need raises interesting questions about how the computer should develop its suggestions. (These issues are discussed further under Concept 5.)

Concept 4. Facilitate communication and cooperation by designing a system that can infer the planner's current goals.

In selecting a flight amendment to deal with some problem, the solution space that could be searched is often quite large. If the computer can determine what the planner is trying to accomplish, it can begin this search on its own.

One example of such aiding involves avoiding bad weather. When the planner sketches a solution on the map display in order to explore a route south of some storm activity (as described under Concept 3), FLIGHT PLANNER infers the planner's goal and automatically begins searching for alternative solutions (e.g., going north of the storm, or flying above the storm). If a promising alternative solution is found, this is displayed to the planner for consideration.

Concept 5. Be sure there is a clear, easy to understand conceptual model for controlling and understanding the computer's processing.

The assumption behind building cooperative systems is that two "heads" are better than one, especially when one of them is only a computer. This raises some interesting questions:

1. Should we try to design the computer so that it thinks "like" people do?
2. How do we ensure that the human planner and the computer system have
   the same goals and priorities?
3. How do we design the system to induce the human planner to play an active role in planning rather than relying on the computer to do all the work?

Lehner and Zirk (1987) present data suggesting that computers need not think exactly like their human partners. Indeed, they found that best performance occurred when the computer did not use the same reasoning processes. A necessary condition for this result, however, was that the human partner be able to understand how the computer arrived at its conclusions.

Several flight planning systems have been developed that use optimization techniques to find the "best" plan for a given situation (Sorensen, Waters and Patmore 1983). To use such systems, the planner must assign weights to different factors such as fuel consumption and tardiness. This is certainly different from the way humans reason about flight planning (Galdes and Smith, 1990). It is also, however, difficult for humans to understand the underlying reasoning. We are consequently investigating the development of "cognitive interfaces" to such optimization systems.

At one extreme is a system that simply finds the "best" route in terms of a single objective, such as fuel consumption or arrival time. The human planner is then forced to play a very active role, looking at other factors such as turbulence.

At the other extreme is a system where the human planner can set up constraints for the flight, such as:

1. Minimum acceptable remaining fuel;
2. Earliest acceptable arrival time;
3. Latest acceptable arrival time;
4. Maximum acceptable turbulence level;
5. Minimum clearance from thunderstorms.

Such constraint setting is more compatible with normal human planning considerations (Galdes and Smith, 1990) than asking the person to weight the relative importance of different factors. There is still, however, a need to support independent planning by the person. What if, for instance, the plane has pressurization problems and can't climb to its normal altitudes? What if the passengers have just had lunch? What if the nearest accessible alternate airport is further away than originally planned because of bad weather?

Thus, we are using FLIGHT PLANNER to study the use of optimization algorithms and the design of cognitive interfaces to these algorithms. We are also, however, studying ways to support independent human planning, and studying ways to ensure that such planning will actually occur in a timely fashion.

Finding Alternative Destinations. Similar issues arise in developing aids to help find a new destination. One approach is to have the system generate a "best" alternative. This approach, however, assumes that the computer knows what "best" is for the particular situation. In some cases this will be determined by the time required to get there (as in an acute medical emergency). In other cases, it may be determined by a combination of factors such as the degree of traffic congestion and the availability of passenger connections. In still other cases, it may be determined by the amount of fuel needed to get there. At a minimum, the human planner must know how such a system defines "best", so that he/she will know when to ignore its recommendations. (Even with such knowledge, though, the human planner may become overreliant on the system and fail to note a problem with its recommendations.)

An alternative design is to develop a system that the human planner can query, asking questions like:

> What airports can this plane reach within an hour?
> What airports can this plane reach with 15,000 pounds of fuel?
> How long will it take to get to ORD?

Such a design ensures that the human planner takes an active role in the problem-solving as he/she must integrate such information in the selection of an alternative destination. It also, of course, increases the human planner's workload.

Concept 6. Create a microworld in which the person can actively explore "what-if" questions and get

useful feedback to help in evaluating alternative plans.

The literature on intelligent tutoring systems discusses the use of computer-supported "microworlds" to allow students to explore (Wenger, 1987). The same concept is supported in FLIGHT PLANNER. The planner can ask questions like: What if I go north around the storm or fly over it? FLIGHT PLANNER provides feedback regarding fuel consumption, arrival times and turbulence:

Concept 7. Support a variety of planning "models" to accommodate different situations and people.

In our simulator studies of flight crews, we observed several different planning "models" in use. An effective cooperative system should probably accommodate all of these "models."

Planning Model 1. The most common cause of flight amendments is some localized disturbance that makes the plane's original flight plan undesirable or impossible. Typical causes include:

1. the development of areas of turbulence;
2. the unexpected formation of localized storms;
3. changes in winds at different altitudes;
4. the appearance of other air traffic that prevents planned altitude changes.

Example 1. In our simulator study, the flight crews noted that they were behind schedule and burning up more fuel than expected under the original plan. They concluded that the problem was a headwind that was stronger than expected under their original plan. The crews asked ATC whether there were any reports on winds at other altitudes. They learned that the headwinds were favorable at lower altitudes. They compared the tradeoff between the benefits of the lower headwinds and the cost of flying at the lower altitude, and decided it was preferable to fly at a lower altitude. They requested clearance from ATC to do so.

Example 2. Flight crews encountered light to moderate turbulence. They considered changing altitudes to avoid it, or slowing the plane to reduce its effects. They checked for pilot reports on the likely duration and magnitude of the turbulence at that altitude, and on turbulence levels at other altitudes. The turbulence was reported to be very localized, so they decide to ride it out, slowing down to reduce its effects.

Planning Behavior. Our data indicate that, currently, flight crews generally respond to such localized disturbances by generating solutions that are minor modifications of the original plan. In most cases, the crew doesn't replan the entire remainder of the flight, they simply select an immediate response to the local problem and act on it. They assume that they will be able to find additional minor modifications for the remainder of the flight when the need arises (Suchman, 1987).

Model 1 - Discussion. Three points merit discussion. First, under these circumstances, plans are generated by attempting to make minor modifications to the original flight plan. It is assumed that, because the modifications are small, the potential implications for later in the flight do not have to be considered in detail. It is assumed that any later modifications made necessary by the current change will again be minor, and that acceptable modifications will be possible. A second point is that such planning is very decentralized. ATC looks at the local implications in terms of air traffic, but other than that, no one evaluates the effects of the requested amendments on the overall system. No one says "there's been a disturbance, let's now replan everyone's flight" to ensure "optimal" or good overall system performance.

This decentralized approach to planning makes strong assumptions about the "world". It assumes that the flight plans of different planes are not tightly coupled. It assumes small changes in one plane's plan do not usually result in significant disruptions of other plane's plans, or of overall system performance. It also assumes that the "world" generally allows a variety of small changes to be made. Consequently, it is unnecessary to anticipate the availability of future modifications that will be made necessary by the current minor modification. It is assumed that some acceptable modification will always be available to meet future needs.

The third point is that, at present, such localized planning is accomplished in one of two ways. The first method can be characterized as a simple forward search with a short planning horizon. The pilot looks at the immediately available alternatives (changes in altitude, vectoring around the storm or turbulence, slowing down to reduce the effects of turbulence, etc.) and picks the one that seems to best solve his/her immediate problem. The second method is somewhat analogous to case-based reasoning (Riesbeck and Schank, 1987), except that the pilots access a broader "institutional" memory. They ask ATC whether any other pilots have already found a solution to the immediate problem and then make use of that solution (with minor modifications as needed).

Our present design of FLIGHT PLANNER currently supports such decentralized, localized planning. The planner can use the map display to find a set of waypoints that take the plane around a storm. The planner can also view the detailed spreadsheet and look at fuel consumption, winds and turbulence for the next flight segment in order to decide whether to change altitude. It would also be possible to support the case-based reasoning solution by providing the planner with access to already tried localized solutions that have been successful. The planner could then make minor modifications to these successful plans,

Planning Model 2. Under Planning Model 1, the planner doesn't worry too much about a complete path to his/her destination. He/she simply finds an amendment that solves the immediate problem and assumes that the remainder of the solution can be worked out when the time comes.

We also saw cases where the pilots in our simulator study worked out the entire flight plan after proposing an amendment. In such cases, planning was again very decentralized. No one asked: What's best for the whole system? ATC did, to some extent, look at the interactions among planes and put constraints on the solutions. The flight crew simply searched for a solution for their own plane alone that met these constraints.

There are several ways in which a flight planning aid could support such planning. The first would be to provide the raw data and calculations (winds, turbulence, fuel consumption, etc.) necessary for the human planner to work out a complete solution using forward search methods. The second would again mimic case-based reasoning approaches, borrowing from already generated solutions used by other planes.

The third approach mimics current human-to-human interactions. In our simulation studies, we sometimes saw pilots develop fairly abstract plans and then let ATC or Dispatch work out the details. They would say things like:

> "Can you find us a route north of this storm? or
> "We need a new destination airport."

By supporting planning at different levels of abstraction, our testbed mimics some aspects of this human-to-human interaction. Additional features worth considering based on this model, however, include allowing the human planner to specify a goal or constraint (such as "find a route that gets me to my destination within 10 minutes of my scheduled arrival time" or "find me an alternate destination" or "find a good airport that I can reach within 30 minutes" or "find an airport that I can reach and still have adequate holding fuel.")

Planning Model 2 - Discussion. Planning Model 2 has two important characteristics. First, like Planning Model 1, the planner doesn't worry (too much) about finding global solutions that lead to good overall solutions for all of the air traffic. Second, unlike Planning Model 1, the planner works out the entire remainder of the flight. He/she uses a much longer planning horizon.

Finally, as discussed above, our simulation data suggests that pilots currently use a variety of solutions to generate such plans. They use forward search methods; they use case-based reasoning; they plan at higher levels of abstraction and then offload planning to another agent by merely specifying a goal or constraint. All of these methods have potentially important implications for building computer aids.

Planning Model 3. Planning Models 1 and 2 involved looking for solutions from a decentralized perspective. The planner (the flight crew in this case) looked for a plan that was good for him/her without directly considering whether that plan was good from a global perspective. (The global perspective was still partially considered by ATC when deciding whether to approve a requested change in altitude, etc.)

A third planning model that we have seen in use involves explicitly considering the bigger picture. Such planning is currently done by ATC and Dispatch. This model is typically invoked when there is some large, systemic disturbance (a line of thunderstorms, airport closings, etc.). In such a case, ATC and Dispatch look for broader solutions that consider the overall implications for all of the air traffic (or at least that airline's air traffic). At present, this global planning involves both elements of cooperation and competition. Dispatch would like to get the best solutions for his/her airline. ATC would like to find good overall solutions.

From the flight crew's perspective, such planning often takes the form of case-based reasoning. The crew is informed that ATC has developed a preferred alternate plan for planes along that path, or that Dispatch has a recommendation. The crew then evaluates this plan to ensure that it is acceptable to them.

Concept 7 - Discussion. Above, we describe a variety of planning "models" and methods that we have observed in use under current circumstances. These observations are of considerable importance, as it is likely that an effective cooperative systems should support such alternative "models" and planning methods.

Concept 8. **Use graphics to enhance perceptual processes, helping the planner to "see" the important patterns instead of making him/her laboriously "reason" about the data in order to infer their presence.**

The attention literature makes a distinction between automatic recognition processes and controlled processes. Larkin and Simon (1987) suggest this concept can be fruitfully applied to designing aids for problem-solving.

The most interesting application of this concept to flight planning is with the map display. By allowing the planner to view the plane moving along its route, viewing concomitant changes in the weather, the planner may find it much easier to judge trends and note important patterns.

The detailed spreadsheet illustrates another simple application of this concept. By embedding graphics identifying the current flight plan, "optimal" plan and maximum altitudes into the spreadsheet, it should be much easier for the planner to identify pertinent data and make comparisons at different altitudes. We may also graphically embed cloud TOPS into the spreadsheet at some point.

Concept 9. When using graphics, provide a "natural" mapping between the features of the display and the corresponding concepts or real-world objects.

The map display is an obvious application of this concept. The detailed spreadsheet is also consistent with it, however. The spreadsheet depicts the horizontal movement along jetways as a horizontal sequence of cells on the spreadsheet. Each successive column represents the next waypoint or jet route in sequence. (An interesting conflict arises, though, when the plane is flying east to west. Should the sequence on the spreadsheet now go from right to left to be consistent with the orientation of the map display?)

The altitude information at the bottom of the spreadsheet is also consistent with this principle. Higher altitudes for a flight segment are represented as higher cells in the spreadsheet.

There is also another inconsistency with this principle. The length of flight segments is not reflected at all in the graphics on the detailed spreadsheet. All spreadsheet columns are equally wide, even though the flight segments they represent differ in length. We have experimented with displays where segment lengths were drawn to scale. Segment lengths differ greatly, however, and our judgment was that it would be better to tradeoff in favor of compactness of the display (allowing the planner to see more flight segments at a time) rather than having pictorial realism.

Concept 10. Consider distributing the problem-solving to simplify the tasks for individual participants.

At present, there are several parties involved in flight planning. The flight crew plays a major role in detecting problems that require replanning. The flight crew also does much of the replanning. ATC sometimes generates some of the details of a plan, but often ATC plays a reactive role, telling the flight crew whether an amendment they have proposed is feasible given other air traffic.

Similarly, Dispatch often plays a reactive role, relying on the flight crew to detect a problem and to suggest a solution.

These roles depend very much on the time-criticality of the problem and its nature. Dispatch is more likely to play a major role in selecting an alternative destination, for instance, than in proposing a change in altitude to avoid turbulence.

It is clear, then, that there is currently a decomposition of flight planning activities that allows different parties to deal with different aspects of the flight planning problem. Such task decompositions need to be considered when deciding who should have access to what information and computer aids.

Concept 11. Consider including redundancy in a distributed problem-solving environment to increase the likelihood that good solutions will not be overlooked and that bad solutions will not be accepted.

In addition to reducing the cognitive load by distributing tasks among different parties, such shared problem-solving may benefit from intentional or chance occurrences of redundancy. Dispatch, for example, may notice that a flight amendment proposed by the flight crew leaves very little holding fuel and recommend finding an alternative plan.

In designing the planning environment, we may want to use computers and advanced communication capabilities to enhance such intended and incidental redundancy. There may be data and information that we want to deliberately present to multiple parties. This may include presenting the computer's conclusions, explorations and warnings to both the flight crew and Dispatch (and in some cases, to ATC as well).

The literature on human error discusses such things as the generation of false assumptions (Fraser, Smith and Smith, 1990; Smith, Giffin, Rockwell and Thomas, 1986), and fixations on incorrect hypotheses or unwise solutions. In our simulation study we saw one example of such behavior. One crew appeared to fixate on Toledo as an alternate destination after Detroit was closed. Initially, it appeared to be a reasonable alternative, but given the questionable weather in the area and the progressively lower fuel levels, it was a very dubious choice to commit to while over Gopher. The crew never asked: Do we have enough fuel to go elsewhere if the weather at Toledo turns bad (or if air traffic congestion develops)? Similarly, we saw several cases where flight crews failed to consider the implications of certain events (being held at a lower than planned altitude) or actions (flying faster than normal cruise speeds). Appropriate aids to enhance distributed problem-solving might help reduce such "errors."

Concept 12. Design assuming that novel situations will arise that will make invalid certain inferences and conclusions made by the computer system.

It is clear that knowledge-based systems and optimization programs have limited scope. It is quite probable that situations will arise that were not anticipated by the system designers.

One solution is to provide the computer system with explicit error detectors (Smith, Smith, Svirbely, Miller, Glades, Fraser, Blazina and Kennedy, in press) and with metaknowledge. To the extent that the computer knows what it does and doesn't know, it will be better able to detect situations where it is "over its head."

This solution simply reduces the likelihood that the computer will unknowingly generate a questionable plan. There is still the likelihood that the system designers will leave out important metaknowledge to detect some novel situations.

A second solution, therefore, is to keep people actively engaged in the planning activities, and to attempt to ensure that they consider important data as well as recommendations by the computer (or another person). This requires careful consideration of the roles of various agents (human and computer) as well as the design and distribution of data displays.

Concept 13. **Try to predict the errors that components of the system, individually or jointly, could make. Try to design the overall system to prevent errors. Equally important, try to design the system so that errors (including those that haven't been predicted) are likely to be caught or, failing that, so that their impacts are not serious.**

In our interviews and in our simulator studies, the most serious situations seem to result from a combination of three factors:

1. Using a short planning-horizon to solve some immediate problem (thus failing to consider long-run implications);
2. Failing to discard the current plan early enough, while there are still many alternative options available;
3. Experiencing the occurrence of a series of events that, taken together, seriously threaten the plane's safety, even though each one alone would normally be a minor problem.

Under Concept 7, we describe a model of planning in which planning is very localized, in which the pilot finds a solution to the immediate problem without considering in detail the implications for later in the flight. This form of planning assumes a "friendly" world, where there are numerous alternatives to select from to solve the next step in developing a plan. Under such an assumption, there is no great need to look beyond solving the immediate problem.

In flight planning, the assumption of a "friendly" world is normally quite viable. The plane has reserve fuel, keeping many options open. The plane can land somewhere else if fuel, weather, etc. make this necessary. Finally, the pilot can request priority clearances if the situation is becoming sufficiently difficult, thus gaining additional options.

Occasionally, however, the flight crew finds itself in a less "friendly" world. Based on our interviews, this seems to arise for one of two reasons:

1. The plane encounters a series of problems that require flight amendments and use up extra fuel. The solution to each problem taken alone is quite reasonable, but, taken together, fuel levels get unacceptably low. Thus, by failing to consider a longer planning horizon, and by failing to anticipate potential "worst case" possibilities, the crew ends up in a situation where they have few good options left;
2. The crew "fixates" on their current plan too long, failing to notice that their other options are disappearing (due to low fuel). If the "worst case" arises and they can't complete their current plan, they are in a difficult situation.

Solutions. One solution would be to make the world "friendlier." The obvious (but expensive) way to accomplish this would be to require greater fuel reserves and reduce air traffic levels. A second would be to develop computer aids that help the planner to use a longer planning horizon and to anticipate possible "worst case" situations. A third would be to develop aids that monitor the situation and warn the planner when the number of options is becoming dangerously low. A fourth would be to facilitate distributed planning on the assumption that Dispatch, for example, might be less likely to share a fixation that the crew has developed (or vice versa).

Conclusion

Technological and conceptual advances in the design of knowledge-based systems, in optimization methods and in telecommunications offer powerful tools for improving performance in complex systems. In applying such technologies, however, we must identify the true problems and needs
of the application area, and understand the limitations of the available technologies.

An important conceptual approach to the development of computer-based cognitive tools or aids is to explicitly design systems to enhance cooperative problem-solving. This approach starts with the assumption that, for both economic and technological reasons, there are many areas where complete automation is unlikely to provide an acceptable solution.

Consequently, if we are to make effective use of current computer capabilities, we need to understand how to design cognitive aids that people can work with effectively.

Above, we describe an effort to apply this conceptual approach to the development of FLIGHT PLANNER, an aid for enroute flight planning. As part of the process of building this artifact, we have identifed a number of general design concepts that proved useful in guiding design decisions. These design concepts, discussed and illustrated above, serve to point out possible ways to improve overall system performance by facilitating shared problem-solving.

References

Fraser, J., Smith, P. J. and Smith, J. (1990). A Catalog of Errors. Technical Report #CSEL- 1990-14. Department of Industrial and Systems Engineering, The Ohio State University.

Galdes, D. and Smith, P. J. (1990). A Cognitive Task Analysis of Enroute Flight Planning Activities by Flight Crews. NASA Technical Report (in Press).

Laird, J. E., Newell, A. and Rosenbloom, P. (1987). SOAR: An Architecture for General Intelligence. Artificial Intelligence, 33, 1-64.

Larkin, J. and Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 11, 65-99.

Lehner, P. and Zirk, D. (1987). Cognitive factors in user/expert-system interaction. Human Factors, 20, 97-109.

Michie, D. (1986). On Machine Intelligence, 2nd Edition. Chichester: Ellis Horwood Limited.

Norman, D. and Draper, S. (eds). (1986). User Centered System Design. Hillsdale, NJ: Erlbaum.

Reisbeck, C. and Schank, R. (1989). Inside Case-Based Reasoning. Hillsdale, NJ: Erlbaum.

Sacerdoti, E. (1974). Planning in a hierarchy of abstraction spaces. Artificial Intelligence, 5, 115-135.

Smith, P. J., Smith, J., Svirbely, J., Galdes, D., Fraser, J., Rudmann, S., Thomas, D., Miller, T., Blazina, J. and Kennedy, M. (in press). Coping with the complexities of multiple- solution problems: A case study. International Journal of Man-Machine Studies.

Smith, P. J., McCoy, C. E. and Layton, C. (1989). Enroute Flight Path Planning: Cooperative Performance of Flight Crews and Knowledge-Based Systems. Technical Report #CSEL- 1989-8, Department of Industrial and Systems Engineering, The Ohio State University.

Smith, P. J., Giffin, W., Rockwell, T. and Thomas, M. (1988). Modeling fault diagnosis as the activation and use of a frame system. Human Factors, 28(6), 703-716.

Sorensen, J., Waters, M. and Patmore, L. (1983). Computer Programs for Generation and Evaluation of Vertical Flight Patterns. NASA Report #3688.

Suchman, L. (1987). Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge, England: Cambridge University Press.

Wenger, E. (1987). Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Los Altos, CA: Kaufman.

Wilensky, R. (1983). Planning and Understanding. Addison-Wesley: London.