Decision-Making in Time-Critical Situations

Shoichi Masui,1 John McDermott, and Alan Sobel

Department of Computer Science

Carnegie-Mellon University

Pittsburgh, PA 15213

Abstract. There are a variety of tasks in which the amount of lime available to make decisions does not always allow a careful consideration of all oi the options. AIRPLAN is being developed to assist with one such task, managing the launch and recovery of aircraft on a carrier. This paper describes attempts to represent AIRPLAN's knowledge in a way that enables it to provide as much assistance as possible in the time available.

1. Introduction

AIRPLAN is a rule-based system being developed to assist air operations officers with the launch and recovery of aircraft on the carrier U.S.S. Carl Vinson. AIRPLAN's role is to accept raw data about the current situation (eg, the fuel state of an airplane, the weather conditions at a possible divert site), propagate the implications of that data, alert the air operations officers of possible impending problems, and make recommendations for how to resolve those problems. AIRPLAN is implemented in OPS7, a general purpose, rule-based language [Forgy 82], ZOG, a rapid response, large network, menu selection system used for man-machine communication serves as the interface between AIRPLAN and the ship's officers [Robertson 79]. AIRPLAN began to be used in an experimental mode on the Carl Vinson in March of 1983.

The task AIRPLAN assists with has strong time constraints. Though an important objective in developing the system is to provide more adequate advance warning of impending problems, situations can arise in which the amount of time available is insufficient to adequately explore the implications of whatever events have just occurred. Thus one of our primary goals in developing AIRPLAN is to explore strategies that will allow the system to tailor its decision-making assistance to the amount of time available. After briefly describing some of AIRPLAN's current capabilities, we discuss the work we have done on time-constrained reasoning and then identify issues we want to explore further.

2. Airplan's Task

One *of* the primary responsibilities of the air operations officers is to insure that the planes launched from the carrier land safely. These officers monitor the launch and decide on the general recovery stategies for planes once the planes have returned from their missions; they are not responsible for mission planning or for close in air traffic control. Though the air operations officers must deal with a variety of problems, the most common problem is insufficient fuel; a fuel problem may arise, for example, if a pilot is unable to land his plane on his first attempt or if a plane has to stay in the air longer than expected because of a fouled deck. There are three options available if a plane is low on fuel: (1) the plane may be able to be landed on the carrier immediately, (2) the plane may be able to be refueled in the air by a tanker, and (3) the plane may be able to be diverted to an airport on land. The air operations officers currently use what is called a greaseboard (a large, transparent "blackboard") as their information source. Enlisted men stand behind the board and continuously update information about the ship, the planes, and the possible divert sites as they receive reports from a variety of sources.

AIRPLAN's function is to provide assistance to the air operations officers by identifying impending problems and recommending ways of avoiding those problems or at least of making their consequences less serious. The same information that is written on the greaseboard is entered into AIRPLAN. Some of AIRPLAN's rules simply propagate the implications of new pieces of information. Other rules recognize problems and initiate an analysis of the options available for solving those problems. For example, if a plane

Present address: Systems Development Laboratory. Hitachi, Ltd., Kawasaki 215, Japan

2This research was sponsored by *the* Defense *Advanced* Research Projects Agency (DOD). ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory under Contract F33615 78 C 1551. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

queued for landing reports an unexpectedly low fuel state, several options are ordinarily open to the air operations officers once they have recognized that the fuel state is lower than it should be. What they need to determine, for the particular situation at hand, is which option is best. AIRPLAN assists with this analysis by recognizing interrelationships among the problems it has identified, and by considering the implications of events which, if they wore to occur, would create further problems.

3. Airplan's Dealings with Time

Because air operations officers are frequently confronted by problems that must be solved very quickly, they must be continuously sensitive to time issues as they attempt to resolve those problems. The officers are faced with a stream of event reports, any of which may signal an impending problem. If the frequency of the reports is such that it is not possible to consider all of the implications of each report, the officers must allocate an appropriate amount of time to each. Furthermore, since the problems which arise may be equally urgent or may be interrelated, the officers cannot simply attend to the reports one at a time. Since AIRPLAN's role is to assist the air operations officers and since its assistance is potentially most valuable in situations in which too much is happening to make adequate analysis possible (either by the air operations officers *or* by AIRPLAN), AIRPLAN, too, must be able to manage its use of time.

Over the past year, we have explored a variety of ways of enabling AIRPLAN to deal with time constraints. The major difficulty we encountered was in finding a decomposition of the domain knowledge that would allow AIRPLAN to attend first to the most significant aspects of a problem and then, if time remained, to extend its analysis on the basis of its initial conclusions. In the current version of AIRPLAN, knowledge of the first sort recognizes the difference between "possible" and "bad" options; essentially, this knowledge enables AIRPLAN to provide information to the air operations officers which helps them assess the seriousness of a situation. The knowledge which refines the initial conclusions characterizes options as "good", "ok", "poor", or "impossible"; this knowledge enables AIRPLAN to provide recommendations about how to resolve problems. The rules that do the rough characterizations decide whether an option is possible by determining whether each of the factors that need to be considered is above a threshhold value; if an option is not recognized as possible, it is assumed to be bad. The rules that refine these initial conclusions examine each of the factors more closely and take their significance and their relative values into account. On the basis of this additional analysis, possible options are characterized as either good or ok; bad options may become ok as well as poor or impossible.

Given this decomposition, AIRPLAN is able to provide four levels of assistance to the air operations officers:

1. Display raw data.
2. Identify problems and roughly characterize the possible solutions.
3. Refine the characterization of possible solutions.
4. Identify problems that might possibly arise in the future.

As soon as AIRPLAN receives a report, it updates its display so that the officers always have access to the most recent data about planes, the weather, etc. It then determines the implications of that report and if it discovers a problem, characterizes the options as possible or bad. If it has no backlog of unanalyzed reports, it continues with its analysis and the result is a set of recommendations (with justifications if desired). Whenever AIRPLAN has no reports to consider, it generates hypothetical future events (of the sort that frequently occur and cause problems) and indicates what options would be available if such events were to occur.

The structure which supports these four levels of assistance is quite simple. AIRPLAN has several demon rules which recognize new reports. As soon as a new report is received, one of these rules updates the display, notes that a new report requires analysis, and on the basis of the type of information in the report, notes its probable urgency. AIRPLAN then returns to whatever task it was previously engaged in. When it finishes a task (which may have been interrupted several times), it considers the most urgent unanalyzed report; if the information in that report implies that there is a problem, AIRPLAN does the initial characterization of options available. At that point, it determines whether continuing will result in some of the as yet unanalyzed reports being ignored for too long a period. It then either continues with its analysis of the current report or displays its rough characterization and turns its attention to another unanalyzed report. If all pending reports have been partially analyzed, it continues with the analysis of the oldest one. Whenever no reports are pending, it creates a pseudo-report describing some event which, if it were to happen, might create a problem; it then analyzes this report just as if it were a real report.

For a system to be able to effectively manage its time resources, it must be able, while engaged in some task, to interrupt itself when it decides that it has spent as much time as it can afford. It may appear that we have ignored this problem since AIRPLAN can turn its attention to a new task (eg, continue to refine a characterization of options available) only after it has completed some other task (eg, roughly characterize the options). We believe, however, that the notion of "interrupt itself" is mostly a grain size issue; that is, in some real sense, AIRPLAN does interrupt itself when it has finished with its initial characterization of options, but that fact has been masked by our description of the process. What actually happens when AIRPLAN finishes its rough characterization is that a demon recognizes that a state has been reached in which it may be appropriate to turn to some other task; the demon interrupts if there is a backlog of reports. The important point is that AIRPLAN can turn to another task only when it is in a state in which its results to that point are not subject to misinterpretation. The difficulty we faced in finding an appropriate decomposition of AIRPLAN's knowledge was finding a decomposition that allowed such intermediate stable states. We believe the inadequacy of AIRPLAN's time managment abilities is primarily due to the too large grain size of the current decomposition. In order to make AIRPLAN a more effective assistant, we must find ways of further decomposing its knowledge so that the intervals between stable states become much smaller.

Acknowledgements

Many people have provided help with the design and development of AIRPLAN. In particular we want to recognize the contributions of Rob Akscyn, Charles Forgy, Ouei-len Lee, Dick Martin, Don McCracken, Kevin Moore, and Paul Fischbeck.

References

[Forgy 82] Forgy, C. L.
OPS7 User's Manual.
Technical Report, Carnegie-Mellon University, Department of Computer Science, 1982.

[Robertson 79] Robertson, G., D. McCracken, and A. Newell.
The ZOG approach to man-machine communication.
Technical Report, Carnegie-Mellon University, Department of Computer Science, 1979.