ARTIFICIAL INTELLIGENCE AND HUMAN ERROR PREVENTION:
A COMPUTER AIDED DECISION MAKING APPROACH

Six Month Interim Report
Covering the Period November 3, 1977
to May 3, 1978

Principal Investigator: R. T. Chien
Contributors: L. Hollaar
L. Peterson
C. Ross
W. Brew
D. Chen
R. Morishige
Y. Pan

This work is sponsored by the United States Department of Transportation
under Contract DOT-08-80020

July 31, 1978
# Table of Contents

1. INTRODUCTION ...................................................................................................................... 1

2. KNOWLEDGE BASE REQUIREMENTS FOR THE SECURE SYSTEM .............................................. 4
   2.1. Results of the Crew Member Interviews ................................................................. 4
   2.2. Knowledge Base Requirements for an Intelligent Monitor ..................................... 9
       2.2.1. Determination of the Normal State ................................................................. 10
       2.2.2. Diagnosis of Probable Cause ........................................................................ 11
       2.2.3. Automatic Checklist ..................................................................................... 12
   2.3. Knowledge Base Requirements for an Instrument Verification Subsystem ......... 15
       2.3.1. Exhaust Gas Temperature (EGT) ................................................................ 15
       2.3.2. Fuel Quantity ................................................................................................. 16
       2.3.3. Hydraulic Pressure ....................................................................................... 16
       2.3.4. Inertial Platform ......................................................................................... 17
       2.3.5. Angle of Attack/Pilot-Static Inputs ............................................................ 17
   2.4. Knowledge Base Requirements for a Display Priority Resolution Subsystem .... 17
   2.5. Knowledge Base Requirements for a Planning System .......................................... 20

3. CONCEPTUAL STRUCTURE OF THE SECURE SYSTEM ................................................................ 24
   3.1. The Knowledge Base ............................................................................................... 28
   3.2. The Instrument Verification Subsystem .................................................................. 30
       3.2.1. The Detection of Inconsistencies from Multiple Instrument Outputs .......... 32
       3.2.2. The Isolation of Faulty Instruments by Reasoning ........................................ 32
       3.2.3. Confirmation by Related Instruments ......................................................... 35
       3.2.4. The Collection of Time Related Information ............................................... 36
       3.2.5. Diagnosis by Experimentation .................................................................... 37
       3.2.6. The Implementation of the Instrument Verification System ...................... 39
           3.2.6.1. Knowledge About Flight Parameters .................................................. 39
           3.2.6.2. Knowledge About Flight Instruments ................................................. 40
           3.2.6.3. Knowledge About Ground Navigation Stations ............................... 41
           3.2.6.4. Tolerance Propagation ....................................................................... 42
   3.3. The Script Based Monitor ........................................................................................ 43
   3.4. The Aircraft Simulator ......................................................................................... 53

4. DISCUSSION AND FUTURE PLANS ...................................................................................... 57

REFERENCES ................................................................................................................................. 58
1 INTRODUCTION

This report describes the progress made at the Coordinated Science Laboratory in developing the conceptual machinery of a system for enhancing the safety of commercial flight operations. This system, placed on board an airliner, would serve as an additional crew member by performing continuous monitoring of all systems and would perform tasks delegated it by the pilot during periods of heavy workload. Since the purpose of this system is to provide Safety Enhancement by Computer REasoning, it has been named the SECURE system.

A study was made to delineate the functions for the SECURE system and to provide an initial organization [1]. The emphasis is in intelligent monitoring. SECURE interprets the instrument outputs with respect to the flight phase and determines whether a system is operating properly or whether an instrument is defective.

The study also suggests that the SECURE system be constructed in a problem solving environment. This would allow the system to develop plans for recovering from system failures or for operating in degraded modes. These plans could then be presented to the pilot as suggestions which he may either accept or reject based on his knowledge of the situation.

Finally, the study suggests that the SECURE system should alert the pilot to malfunctions in an intelligent manner. For instance, on a clear day, during takeoff, if the radar fails SECURE should not immediately report this fact to the pilot because it will be more a distraction than a help. It is much better to notify the pilot when the aircraft is airborne and the pilot's workload reduced. On the other hand, if visibility were low on takeoff due to
fog or rain, a faulty radar system should be reported immediately to the pilot who may then decide to abort the takeoff.

In deciding what information to present to the pilot during such times SECURE must have knowledge of the relationships between the particular flight phase and the correct operation of each aircraft system. In determining the operational criticality of these systems, SECURE must take into account such factors as weather conditions, runway conditions and the pilot's workload. A way for SECURE to then decide what information to display would be to create a list of priorities for the displaying of information about the instruments and aircraft systems. The priorities of this display information would be updated as each new phase of the flight is entered. SECURE then would display information whose priority is greater than a threshold value which is set by the pilot or which is established by some consensus of pilots.

In order for the SECURE system to establish display priorities it must have considerable knowledge about flight procedures for each phase of the flight. For intelligent monitoring it must have knowledge of the flight profile, flight dynamics and the operating ranges of each system for each phase of the flight. In detecting and diagnosing failures, and in planning recoveries from failures and flight deviations, the SECURE system must have knowledge of the function of all aircraft systems so that it can give a description of the fault condition to the degree of detail desired by the pilot. In general the SECURE system must have access to a large body of knowledge which contains information about all the aircraft systems, about energy management, about the flight phases, about flight procedures, about failure recovery procedures and other information to aid the monitoring, planning and pilot alerting functions.
Hence a knowledge base is necessary for the successful operation of the SECURE system. Much of the work during the first six months of this project revolved about how to implement such a knowledge base. Three problems are inherent in establishing the knowledge base, namely:

1. To determine precisely what information the knowledge base should contain.
2. To determine how this information should be represented.
3. To determine how this information is to be used for the various monitoring and planning tasks.

The following pages of this report describe the efforts to establish the knowledge base, and the concomitant development of the conceptual framework for the SECURE system. Section 2 describes the results of the investigations to date in the establishing of the knowledge base for the SECURE system. Section 3 then describes the efforts to implement the SECURE system. Finally, Section 4 discusses the plans for the second half of the project.
2 KNOWLEDGE BASE REQUIREMENTS FOR THE SECURE SYSTEM

As part of the program for determining the information content of the knowledge base, three separate interviews were undertaken to learn from commercial airline crew members their needs from the SECURE system. These interviews have greatly influenced the planning of what functions SECURE should have, and consequently, the various types of information which should be included in the knowledge base. The three interviews were held with multiple jet engine aircraft crew members including a National Guard pilot, a flight engineer from American Airlines and six senior captains who fly wide body aircraft for United Airlines. It was generally agreed that the SECURE system should aid the pilot and crew with monitoring tasks, it should verify the correctness of instrument readings, it should perform some automatic checklisting, it should propose failure recovery procedures and it should inform the pilot of difficulties at appropriate times. The following sections discuss the results of the interviews and describe the types of knowledge the SECURE system needs in order to accomplish these functions.

2.1 Results of the Crew Member Interviews

It was clearly recognized by all those interviewed that there are many non-critical flight tasks and functions which a computer can perform more quickly, accurately, and efficiently than people. On a general level, these include instrument monitoring and record keeping, making routine calculations of weather, energy-management, and navigational data, the storage of emergency and special procedures and aircraft specifications, and their retrieval on demand. These features can be implemented by a computer system in a
straightforward manner so that, for instance, the flight engineer can be relieved of much tedious and time-consuming instrument watching. Thus, he has more freedom to execute higher level functions such as watching for traffic. Accurate and permanent records of the performance of all aircraft components will become available, simplifying maintenance and forestalling malfunctions. The accuracy and quantity of weather data will be greatly increased, improving the generation of flight plans, enhancing safety, and increasing fuel economy. A substantial beginning can be made to satisfy the universally expressed need for assistance and information when entering a holding pattern. In addition, procedures required for dealing with emergencies will be more quickly and easily available whenever needed. This will help to counterbalance any tendencies of the crew to overreact during emergencies which occur very infrequently because of the increased reliability of aircraft.

However, many additional functions for which a strong desire was expressed, the elimination of false alarms, the lessening the workload of the crew in critical situations, the diagnosis of the causes of simple and multiple failures, and the selection and ranking by priority of emergency and error-recovery procedures, require a considerably more sophisticated approach. It would appear that the airborne computer system needs to be intelligent, that is, to perform in a fashion which would be considered to require intelligence if done by a person. It is here that the concepts of artificial intelligence are most relevant, and they will be extensively applied in the design of this system. It has always been the intention of the designers to equip the system with program intelligence, and now, after these interviews, it is clear that such a system would be enthusiastically welcomed by flight professionals also.
For example, an intelligent monitor could do much more than passively observe and record. It would be able to extract from sensor data the same pragmatic information as the human crew does now, and do it faster, more accurately, without fatigue, and in the presence of obscuring complications. It would communicate conclusions and specific, well-organized recommendations to the crew, not merely the raw, undigested data. It could make use of automatic tie-in of warning and recovery, or of multipurpose displays, or a combination of both. In order to sound only true alarms and to ensure highly reliable responses, it should be sensitive to the phase of the flight and to the state of the aircraft. Simple fixed priority schemes are insufficient. For a discussion of tools and methods and for more details about the functional requirements of an intelligent airborne computer system, the Coordinated Science Laboratory report of Morishige [1] should be consulted.

The need to update the data, especially weather and navigational data, stored in the computer, demonstrates the eventual need for a data link between the airborne computer system and ground based knowledge sources. The airborne system clearly needs access to at least the same information that is used by the computer generating flight plans. Topographic information about the terrain over which the aircraft flies, the actual flight plan, the minimum equipment list (MEL), and the configuration deviation list (CDL) can also be supplied via this link. In the design of the system an implementation of this data link capability will be assumed, for without it, many of the most wanted features cannot be provided. However, the system will still be able to function without this link, though in a less than optimal fashion.
All of the persons interviewed wanted to greatly increase the amount of information that is directly available on board the aircraft. Hence, in addition to the real-time data from sensors and the data link, the airborne computer system will need to have a large knowledge base consisting of aircraft specifications, special and emergency procedures, and other less easily categorized knowledge of flight and flying. Even with special preformulated procedures on board and the ability to determine when they are needed, not all situations can be foreseen. The system will need to possess another feature of intelligence - the capability of generating new plans and procedures to deal with unexpected occurrences. It will also have to be able to admit failure and there may be times when it will be the system, not the human crew, which needs help. The system will always remain open to the direction and advice of the crew, and will interrupt its own activity, unless that is determined to be extremely critical, to aid them whenever requested.

The computer system will be designed to operate in the presence of aircraft malfunctions and deficiencies. This is particularly important if a problem develops in flight, rather than being known in advance, as the crew and the system will not have the advance notice of the MEL or CDL. With this capability, the absence or failure of the data link, the inertial navigation system, or any other subsystem will simply mean that a different set of system resources will be called into play. Failure of power to the computer would be a very serious malfunction. The system needs to fail gracefully, and to resume operation with only minimal assistance when power is restored. As flying without the computer system operational is a clear possibility, the crew must retain all of their present competence.
These interviews, especially the one with the United Airlines captains, were highly productive of ideas and information. This kind of close cooperation between the research group and the potential users is needed to produce a truly workable system, namely, an intelligent airborne computer system which is trustworthy, cost-effective, and convenient, and is not a crutch. As this study proceeds, the system designers will continue to have interviews with flight crews, maintaining a close liaison with the operational environment for the mutual benefit of both sides. It would be useful to follow the O'Hare interview of senior Captains with an interview of younger crew members, especially those with an background in Engineering, in order to obtain another point of view on the situations already discussed.

More detailed information is needed concerning actual emergencies and other situations which entail postponement of routine tasks. There are several types of useful information in this area. One type concerns the circumstances which give rise to these situations and the conditions under which they are likely to occur. This knowledge, which can be obtained from further interviews and official NSTB accident reports, will help the system to prevent and forestall difficulties. A second kind of information relates to what actually happens to the aircraft systems in the most common emergencies. This data can be used as a basis for procedures that analyze the cause, or causes, of problems. Then there are the procedures and methods actually used in practice to cope with these situations. The procedures contained in airline training manuals and films are a source being tapped for knowledge in this area, but the observation of actual flight simulator sessions would be a very valuable experience. In these, one has the opportunity to observe directly what pilots really do, not just what they claim they do, under conditions of high workload and stress.
2.2 **Knowledge Base Requirements for an Intelligent Monitor**

The primary purpose of the monitor is to provide continuous, real-time information about the state of the aircraft and its systems. The use of a computer monitor can significantly reduce pilot workload by relieving him of the tedious task of monitoring numerous instrument readings and mentally integrating the raw data to determine the overall systems performance. For example, engine performance is now monitored by checking and correlating the indications for EGT, RPM, fuel flow, nozzle position, EPR, etc. Normally, it would be much easier for the pilot to monitor a single thrust reading. The complete set of parameters would be available to the pilot on request or automatically when needed.

It is generally acknowledged that certain phases of flight are characterized by a high workload for the pilot and crew. During high workload periods, the quality and quantity of the normal monitoring is likely to be diminished. Deviations are more likely to go undetected for longer periods of time. An unnoticed deviation can frequently compound the hazard of a high workload situation. The monitor would not be affected by the increased workload since it will be designed to have excess capacity over peak load situations. Nor would the monitor be affected by typical human failings such as boredom, fatigue, or fixation.

The intelligent monitor would require several supporting functions. The capabilities for the monitor should include the following:

1. Verification of sensor output.
2. Detection of deviations.
3. Diagnosis of a common cause for related sensor indications.
4. Reporting diagnosis to the executive for priority resolution.
5. Automatic checklist presentation.
6. Generation of a history of deviations and significant events for later analysis.
7. Real-time response
8. Degraded mode operation

The basic sequence of operations for the monitor should probably follow the steps below.

1. Sensors provide direct information about the state of the aircraft and its systems. All significant measurable parameters are sensed.
2. The sensor readings are verified for logical consistency. Any sensor indication which is not consistent is given a lower plausibility and is so reported.
3. The verified sensor readings are compared against a context sensitive model for agreement.
4. A diagnosis is made based upon the deviations to determine if there is a probable common cause.
5. Any diagnosis and the supporting evidence (sensor indications) are reported to the executive for relay to the pilot and recorded for later analysis.

2.2.1 Determination of the Normal State

The normal state of a system may depend upon the context of the aircraft. The phase-based context is often reflected in the crew checklist where different modes are specified for different phases of flight. For example, during landing and takeoff the position of the landing gear should be down, but during cruise, the position should be up and locked. Besides configuration, other examples of systems where parameters are sensitive to the
phase of flight are pressurization, the fuel system, and navigation. The intelligent computer system can determine the normal state of some parameters by the use of scripts. Flight plan information will also provide reference information on planned fuel quantity, route of flight, cruise speeds, etc. to monitor navigation.

Often the normal state or range of a system will vary with the condition of the aircraft. For example, if there has been a generator malfunction, then the normal state for the generator bus tie relay should be open. Similarly during the transfer of fuel, the affected valves and pumps should be open and on, respectively.

Factors which are external to the aircraft such as weather or terrain may also have an affect upon the normal state. Bad weather often calls for increased safety margins. In gusty winds or suspected wind-shear, it is normal to increase airspeed when maneuvering close to the ground. Braking on a wet runway will not provide the same deceleration as a dry runway. A wet runway will have a considerable affect upon $V_1$ (critical engine failure speed), which may affect the recommended procedures during the loss of an engine on takeoff roll.

2.2.2 Diagnosis of Probable Cause

Often a malfunction will cause several sensors to indicate a deviation. It is therefore essential to isolate the cause from the symptoms. A simple example of diagnosis with multiple sensor indications is an engine flameout caused by a malfunctioning fuel boost pump. Within a short span of time, the monitor should detect fuel pressure low, RPM low, EGT low, EPR low, associated
generator output off, and low hydraulic pressure on the associated system. A conventional system would provide the pilot with every sensor indication and depend upon the pilot and his experience to analyze the data and draw the proper conclusion.

The following method for diagnosis uses the cause-effect net shown in Figure 1. The cause-effect net would be represented in the computer with directional links between the nodes [2,3] Failure of the number 2 boost pump with the cross-feed valve closed, will cause fuel pressure to drop, which in turn will retard fuel flow, which in turn will cause combustion to cease. Combustion requires the simultaneous and continuous existence of not only fuel flow, but also air flow and ignition. Without combustion, the engine will stop running, which will cause the number 2 generator and number 2 hydraulic pump to fail. If the number 2 AC bus is hot because it is receiving power from the number 1 generator, and the boost pump switch is on, then there is an inconsistency across the number 2 boost pump node. That is, there is no fuel pressure, even though the pump has electrical power. This inconsistency can be found from any point in the net that is affected by the inconsistency by backing up the links until the inconsistency is found. The result of the diagnosis would be presented with a list of supporting indications or consequences. This type of information will particularly help less experienced pilots cope with unusual situations.

2.2.3 Automatic Checklist

The purpose of the checklist is to insure critical actions are not forgotten. Human limitations necessitate that the checklist procedures be brief and cover only the most important items. Therefore, the omission of any
Figure 1. Cause effect network for part of an aircraft engine.
item could result in a hazardous condition. There are two basic types of checklist actions. One is monitoring and the other is preparation. The former is well handled by the basic monitor. The latter implies more planning information.

The preparation for an anticipated maneuver (e.g. landing) may require several actions. The script can provide information on what actions need to be taken to prepare for a maneuver. The preparations include aircraft preparation such as lowering the landing gear, but more importantly, pilot preparation. The pilot would be positively notified that a specific checklist for a portion of the flight had been successfully completed. For example, prior to the approach to landing the display might appear as depicted in Figure 2.

APPROACH TO FIELD CHECKLIST COMPLETED - NORMAL

1. ILS RWY 25L to Los Angeles Intl
2. APPROACH SPEED = 140 KTS
3. MINIMUMS: (200 - 1/2) DH = 326' MSL

VERIFY APPROACH AND LANDING CLEARANCE

Figure 2. Simulated Checklist Display
This kind of information, if presented at the proper time, can help to keep the pilot ahead of the aircraft. In order to provide this information at the proper time, it is necessary to have flight plan and clearance information as well as flight scripts to provide timing and synchronization for the automatic checklist. It should be emphasized that the information in Figure 2 merely supports the Vertical Situation Display (or the Heads-Up Display) and the Horizontal Situation Display which are the primary sources of information during flight.

2.3 Knowledge Base Requirements for an Instrument Verification Subsystem

Sensor verification is the process of establishing the logical consistency among related sensors. The nature of the consistency will depend upon the physical constraints of the sensor being checked. The following description of aircraft parameters indicates how they are related to other aircraft parameters. This set of parameters is incomplete but serves as starting point for establishing a complete set of parameters in the SECURE system. The relationships thus established will become part of the knowledge base and will allow the respective parameters to be checked for consistency with respect to other aircraft parameters.

2.3.1 Exhaust gas temperature (EGT)

This is a primary indication of thrust. A higher EGT will normally indicate a higher thrust. Related sensors are fuel flow, RPM, EPR (ratio of turbine pressure to inlet pressure), air temperature, etc. Normally the fuel control schedules fuel to the engine based upon throttle position, air
temperature, RPM, EGT, and compressor discharge pressure. Proper operation of
the fuel control depends upon correct sensor data. High EGT is normally
associated with high fuel flow, high EPR, and usually high RPM. If the EGT
trend were to follow the other parameters, then the EGT sensor would probably
be operating properly. If a change in EGT were reported without a
corresponding change in the related parameters, then the EGT sensor would
probably be in error.

2.3.2 Fuel quantity

The quantity of fuel within a particular tank can be checked by
transferring known quantities of fuel into or out of the tank being checked. Also, because the flow rate is usually known, it is possible to integrate to
fuel flow over a period of time to derive a value by which the quantity should
change. The indicated quantity of fuel would also be compared against the
planned quantity from the flight plan.

2.3.3 Hydraulic Pressure

The hydraulic pressure is a measure of the performance of the hydraulic
pumps. Related sensors include hydraulic fluid quantity, hydraulic pump rpm,
fluid temperature, and accumulator pressure. Sensors which directly report the
resultant action of hydraulic pressure are also necessary. These would include
flight control surface position indicators, other hydraulically actuated
mechanism position indicators.
2.3.4 Inertial Platform

The verification of this type of sensor is a problem because it is a reference standard. While direct verification of an accelerometer can only be done by using redundancy, the resulting position information of the inertial navigation system can be directly checked by independent means. Similarly, the attitude information can be verified by checking the aircraft's flight characteristics. For example, if the aircraft is not wings level, it will tend to turn. Pitch can be checked against angle of attack and airspeed.

2.3.5 Angle of Attack/Pitot-static inputs

The relationship between the angle of attack and pitot-static data varies as a function of the forces acting upon the aircraft: lift, drag, weight, and thrust. Lift is a function of angle of attack, airspeed, and configuration. Drag also depends on the same parameters. Thrust is simply the effective thrust of the engines. Weight is the effective weight of the aircraft (takes G-loading into account). The angle of attack and the pitot-static data can be correlated with additional data about the engine thrust, aircraft acceleration, attitude, and aircraft weight.

2.4 Knowledge Base Requirements for a Display Priority Resolution Subsystem

The problem for the priority resolution system is to take a massive amount of raw data and to process it in real time and produce relevant information in a form which is readily usable. Most information need not be presented to the pilot, unless specifically requested. In order to prevent an information glut, only essential information should be provided automatically.
A skilled pilot is able to stay ahead of his aircraft by rapid interpretation of his instruments. From practice, he knows where to look and also what to expect to see. While conventional instrumentation lends itself to this type of interpretation, multipurpose CRT displays do not. The pilot cannot look at a multipurpose display and know a priori what will be there. It could be engine data, navigation data, or even emergency data. There are two distinct interpretations that must be made, first, what the information is and then second, what it means. If the handicap of the first interpretation is to be offset, then the quality of the information displayed must be very high. Clearly, there is more to the problem than format and organization factors. The next section is concerned with the content of the information provided. If the content is not relevant, then no amount of formatting or structure will increase the value of the information.

The priority of any information from the monitor may be variable and depend upon the context of the current situation. The criticality of the loss of cabin pressurization depends upon the altitude of the aircraft. The loss of cabin pressurization at high altitude may be caused by loss of pneumatic pressure for air conditioning, cabin air outflow valve stuck open, or loss of structural integrity. It is clear that the loss of structural integrity of the cabin will require a descent. However, in the case of loss of pneumatic pressure, action other than a descent might restore cabin pressurization. If the loss of cabin pressurization is caused by a loss of pneumatic pressure due to an engine malfunction, then the cabin pressure might be most quickly recovered by correcting the engine problem. In practice, the pilot would probably initiate a descent while working on the engine. If the same problem occurred at low altitude, no pilot action would be necessary and accordingly, the pressurization failure would not warrant a high priority. On the other
hand, some warnings, such as an engine fire indication, should always have high priority.

The criticality of a generator malfunction depends upon the operation of the other generators and the engines. The failure of an engine results in the loss of its generator. The loss of the second generator is always more critical than the loss of the first. For most multi-engine aircraft, a single generator can supply all essential electrical power, but all non-essential electrical equipment must be shutdown to prevent overload and subsequent loss of the single generator. The same reasoning is likely to be true of any system which relies on redundancy for reliability.

An example of changing priorities can be seen during the normal takeoff sequence. Through the very early part of the takeoff roll, any deviation may be reported because the aircraft would be traveling slowly and would have sufficient runway to easily stop. However, during the critical phase of takeoff (from about 50 knots until after takeoff is completed), the executive should suppress any information that is not related to takeoff performance parameters, such as engine thrust, flight control and flight hydraulics, emergency electrical power, landing gear and brakes, and the inertial reference system. The pilot should not be distracted with non-critical information during a critical phase of flight. The information could be provided after the takeoff is completed. Besides, the outright suppression of data, any data that is presented, is ordered in a priority ranking so that the pilot will receive the most critical information first. In the case of multiple emergencies, the system would be able to present up to four malfunctions or deviations at once. This would permit the pilot to change the order of priority as he might deem necessary.
Because all high workload or critical phases of flight are not always the same, the aircrew should receive an indication that a low priority deviation has been detected. The indication could be temporarily ignored with reasonable confidence, but if the pilot or another crew member could afford the time, the suppressed information could be called up for display.

2.5 Knowledge Base Requirements for a Planning System

The objective of the plan generator is to provide procedural information to permit the pilot to cope with abnormal situations in the safest possible manner. This is accomplished by providing the necessary information on a timely basis. The normal sequence of operation for the plan generator would start with a signal from the monitor indicating an abnormal condition and a diagnosis. The plan generator would then provide a canned plan from its standard repertoire or generate a new plan. The primary reason for canned plans is to assure minimum reaction time to abnormal conditions. A canned plan might result in the use of automatic severity abatement procedures to further reduce reaction time under certain well defined situations.

The general guidelines for operation during an abnormal situation applies to every situation in any type of aircraft. These rules are as follows:

1. Maintain aircraft control. This means that the priority of the problem should not override the basic "safety of flight" rules. Aircraft control should not be sacrificed in an attempt to solve a problem.

2. Analyze the situation. The pilot and the monitor must collectively determine the cause of the problem.

3. Take the proper action. The plan generator should come up with a sequence of actions that will correct, or at least ameliorate the situation. Failing that, it might try to generate a plan to cope with the situation until a safe landing can be made.
4. Land as soon as conditions permit.

It would be unreasonable to assume that contingency plans for all possible deviations could be anticipated, much less programmed. For this reason the computer system should possess the capability for generating new plans. Plan generators, planners, or problem-solvers might be categorized into two broad types. There are the broad problem solvers which seek to find a general methodology applicable to a broad class of problems. This approach has met with rather limited success, which is not surprising when the nature of the task is examined. An alternate approach using specialized knowledge bases has had more demonstrable results. Some leading examples of specialist systems are MYCIN [4], DENDRAL [5], and MACSYMA [6].

The ad hoc approach would seem most useful at the present time because the aircraft and the flight domain provide some natural constraints which can simplify the plan generation task. The goal is always the same - to land safely at a suitable airfield. A flight plan and the flight script provide short-range goals for the operation of the aircraft. The possible actions which can be taken are limited to a fixed, though large set which can be controlled, either directly or indirectly, from the cockpit. Many actions, such as repairs, are precluded because of inaccessibility of system components during flight. Hence, redundancy is commonly used in the design of aircraft systems which simplifies the problem solving task. Another constraint is that many controls are binary - either a system or function is ON or OFF. In addition, most of the interactions among the systems of the aircraft have been studied in extreme detail and are well understood. These considerations will
facilitate the \textit{ad hoc} plan generator.

An earlier airborne computer system, CADM [7], which involved the generation of actions to correct abnormal situations was limited in certain respects. While CADM was able to solve a large class of simulated malfunctions, it was not sensitive to changes in context during a normal flight. The CADM error correction procedure was developed by the use of a fixed strategy. CADM assumed that the pilot was always right and did not interact constructively with the pilot during the planning stage. Hence, CADM could not utilize the pilot's knowledge of the aircraft and its systems, of procedures, or of high level goals. Finally, if CADM failed, it did not provide the pilot adequate information about the reasons for failure.

The presence of the pilot is also a consideration that must be incorporated into the design of an intelligent airborne computer system. The pilot, with his experience and training, can be a valuable source of information during periods of abnormal operation. For example, the problem of computer planning in incompletely specified domains [8] has not really been solved. The pilot can provide high level guidance to minimize the possibility of the computer system becoming side-tracked. Pilot inputs could be the specification of goals, resolution of priority, resolution of conflicting information, or the specification of current limitations. While excessive pilot dependence is not desirable, some interaction would be of mutual value to the pilot and the computer system.

The plan generator need not be able to solve every problem in minute detail. It should do an excellent job on most situations, a reasonable job on most of the other jobs, and fail only partially on the remaining few problems. The concept of the use of a partial plan [9] emphasizes the need for
sufficiency rather than detail. It must be emphasized that a partial plan which may not be adequate for computer implementation, may be overly detailed for use by the pilot. This is because the pilot has much higher level thought than a computer system. A partial plan may also permit additional flexibility in a given situation by allowing the pilot to interpret the plan as necessary. For example, an instruction to increase airspeed may be accomplished by a negative change in vertical velocity, by increasing the engine thrust, by changing the configuration, or any combination of the preceding methods.

Partial plans concentrate on a local strategy to revise faulty plans at the fail point. This means that the part of the plan that failed is analyzed to determine and correct the cause of failure and that, at least temporarily, the rest of the plan is preserved. This is usually more efficient than scraping an unsuccessful plan in its entirety. Furthermore, the reuse of old partial plans can be used by the pilot to evaluate the cause of the plan failure. The pilot or the plan generator could call for a change in global strategy when local strategy changes appeared inadequate.

Once a plan is generated, its execution could be simulated by using the cause-effect net. The simulation may expose a fault in the plan or an undesirable side-effect. If no unacceptable side-effects were detected and the plan had the intended effect, then the plan could be implemented by the pilot. A plan to restart an engine with a failed fuel boost pump could be tested by using the net in Figure 1. The simulation would follow the flow of causality and determine if the desired goal was attained. If the goal was not attained, then the net would provide information about where the plan failed by the inconsistency across a node mentioned above. The simulation should increase the success rate as well as confidence in the plan.
3 CONCEPTUAL STRUCTURE OF THE SECURE SYSTEM

The design for the SECURE system has begun to take shape. Its major functions of monitoring, verifying instrument outputs and planning, using knowledge about the aircraft system, flight dynamics and flight phases have been defined. The overall organization of SECURE is shown in Figure 3. The dashed lines indicate data transfers, and the solid lines indicate control paths. Conceptually the Instrument Verification System (IVS) and the Script Based Monitor (SBM) operate independently. The SBM accesses the knowledge base to determine the aircraft status, to determine whether a system is functioning properly and to determine where the aircraft should be. It enters new data into the knowledge base about the nature of failures and deviations it discovers. In addition, it can call for plans to be made for recovery from various failure modes.

In a similar manner the instrument verification system uses knowledge about aircraft systems and instruments to check the consistency among the outputs of various instruments. Inconsistency reports are passed to the SBM and are added to the knowledge base as part of the flight history. The IVS can also request a plan to be made for diagnosing a probable cause of inconsistency.

The Fault Recovery Planner (FRP) is evoked by the IVS and the SBM to generate plans for recovering from system faults and for operating in degraded mode. These plans are not automatically effected, but at an opportune time are presented to the pilot as suggestions which he may utilize or reject. The FRP draws on the knowledge base for information about all phases of the flight, the flight history, the functioning of systems, weather conditions, possible
Figure 3. Organization of the SECURE system.
landing sites, and generally utilizes information which will allow it to make plans which the pilot can accept.

Finally, the aircraft simulator allows the aircraft status vector to be manipulated in order to demonstrate the operation of the SECURE system.

Communication with the pilot occurs through the Priority Resolution System (PRS). The relationship of the PRS to SECURE is shown in Figure 4. The PRS presents information to the pilot about the aircraft status, instrument or system failures, and it presents suggested recovery procedures generated by the planner. It has the added function of determining when and how much unsolicited information should be presented to the pilot. This it decides on the basis of the knowledge it has of flight procedures and the current phase of the flight. The pilot can always request specific information to be displayed in addition to more detailed information about what the PRS is displaying.

The aircraft simulator, and portions of the SECURE system have been implemented. In particular, part of the IVS has been written to determine the consistency of the navigation instruments which compute the aircraft location. Also the scripts of all phases of the flight have been implemented and allow the determination of whether various conditions are normal. For instance, the SBM determines that it is normal for the landing gear to be down during the landing phase. The knowledge base structure is still evolving and currently exists in parts of the IVS and SBM.

Section 3.1 describes the types of information and some of its representations which will comprise the knowledge base. Section 3.2 discusses the current implementation of the instrument verification system and Section
Figure 4. Relationship between Priority Resolution System and SECURE.
3.3 discusses the current implementation of the script based monitor. Finally, section 3.4 discusses the aircraft simulator which has been implemented.

3.1 The Knowledge Base

The contents of the knowledge base for the SECURE system is summarized in Figure 5. The items for which specific information should be maintained is listed in the left hand column. Each of the SECURE systems which accesses the knowledge base is listed at the top of the figure, and the specific knowledge required by these systems is indicated by the check marks.

Although the knowledge base has not been implemented as an identifiable subsection of the SECURE system, we are attempting to organize the types of knowledge so that this is possible. The problem being worked on is that of how to represent the information so that it is generally accessible by more than one system. Since, for instance, the SBM, the IVS, and the FRP all must obtain information about aircraft navigation systems, there shouldn't be three representations for the same bit of information unless something is to be gained by doing so. For instance, a single cause-effect network similar to that of Figure 1, which will describe all the relationships of a functioning engine on a high level, would allow both the IVS and the FRP to do reasoning about the engine system.

Large portions of the knowledge base will be concerned with procedures and factual information such as VOR transmitter locations, checklist procedures, basic fault recovery procedure, and a history of the flight in the form of periodic status reports. It is anticipated that the knowledge base will evolve into a large part of the entire system as the SECURE system
<table>
<thead>
<tr>
<th>KNOWLEDGE OF</th>
<th>SBM</th>
<th>IVS</th>
<th>FSP</th>
<th>PRS</th>
<th>REPRESENTATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hydraulic System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Cause-Effect Networks</td>
</tr>
<tr>
<td>Pneumatic System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Procedural Networks</td>
</tr>
<tr>
<td>Fuel System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flight Control System</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruments</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Tables and Procedures</td>
</tr>
<tr>
<td>Navigation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flight Phases</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Scripts</td>
</tr>
<tr>
<td>Flight Procedures</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault Recovery Phases</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flight Status</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Lists</td>
</tr>
<tr>
<td>Flight History</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Knowledge base requirements for the SECURE system.
The purpose of the instrument verification system is to determine whether the output for any given instrument is reliable. In doing so the IVS attempts to determine what the output should be with respect to other instrument outputs and compares the expected value with the output of the instrument in question.

The IVS is designed to reflect the following considerations. First, whenever the IVS detects an inconsistency in instrument outputs, it should attempt to indicate to the pilot possible sources for the trouble instead of merely giving him a general warning. Secondly, it is not always possible for the IVS to immediately determine whether a change in an instrument output is due to a failure in the instrument (in which case the output of the instrument would be inconsistent with other instruments) or whether it is due to a change in the state of an aircraft system (in which case the output would be consistent with those of the other instruments). This means that a delayed observation may be necessary to provide second order information to decide the consistency of the instruments. Finally, since it is intended that the computer help the pilot by reducing his workload, rather than burdening him with heavy man-machine communication, a troubleshooting strategy that requires pilot cooperation should not be applied unless all other strategies fail.

An approach which utilizes five strategies is being investigated as a solution to verifying instrument outputs. The relations among the five strategies are shown in Figure 6. Inconsistencies are detected by comparing
different estimates of flight parameter values. When a comparison is inconclusive, the verifier can either use related instruments to confirm the consistency of the instruments, or it can initiate a delayed observation to collect extra information. When an inconsistency is noted, the verifier attempts to locate faulty instruments with related-instrument confirmation, delayed observation, or even with the help of the pilot. The following discussion describes these strategies and their implementations.
3.2.1 The Detection of Inconsistencies from Multiple Instrument Outputs

Parameters indicating aircraft status (altitude, location, etc.) can either be read directly from instruments or they can be computed from a combination of instrument outputs. Advanced aircraft are usually equipped with redundant instrumentation, so that one instrument can be checked against the others. Thus, a particular parameter may be computed in several different ways, each based on one set of instrument outputs. This suggests a straightforward strategy to verify instrument outputs. First evaluate each formula for the target parameter and compare the results. If all of these results agree (within a certain tolerance) with each other, it may be assumed that all related instruments are functioning properly and the result is acceptable. On the other hand, if the computed results are not in agreement, there must be at least one malfunction among the related instruments. An attempt is made to then isolate the faulty instrument or instruments.

The example of Figure 7 illustrates the strategy of verifying the instruments of the aircraft navigation system by computing the aircraft location from each set of instruments and comparing these results. If all the results agree with each other, the conclusion is that the navigation systems are functioning properly. Otherwise the inconsistencies are noted and an attempt is made to determine the faulty instrument or instruments.

3.2.2 The Isolation of Faulty Instruments by Reasoning

In verifying instrument outputs, the IVS should give a real-time response if possible. Therefore, any computation and check-up that can be done without further observation should be performed first. For instance, one way of
indicating the faulty instrument is to analyze those source instruments which contribute to the inconsistent results and to issue a candidate list of possibly faulty instruments, ordered by their relative chance of occurrence. If this fails to give a confident answer (or fails to locate the faulty instrument during diagnosis), further information may still be obtained through diagnoses which rely on the delayed observations of certain parameters, may require delays of several seconds before the faulty instrument is located and should only be used as a last resort with the pilot's knowledge.
An approach to determining faulty instruments is to endow the IVS with a "reasoning" capability. An example of the type of reasoning possible is illustrated by Figure 8. Assume the aircraft has the following navigation instruments: INS, VOR1, VOR2 and DME1. If the location reading from INS is consistent with the location computed from VOR1 and VOR2, while the computed result from VOR1 and DME1 is not consistent with the other computations, the IVS will conclude that DME1 is probably faulty.
3.2.3 Confirmation by Related Instruments

Some instrument outputs may not contribute directly to the computation of a parametric value, but can still provide evidence to support or reject the values which have been computed from other instrument outputs. While consistent confirmation from the supporting evidence increases the likelihood of correctly deciding that the instruments in question are functioning properly, an inconsistent report is also useful because it can help in isolating instrument faults.

Figure 9. Using ADF to confirm a location computed from VOR's.
The example illustrated by Figure 9 shows an aircraft with two VOR's and one ADF. Although a single ADF reading cannot lead to an independent estimation of location for comparing with the result computed from two VOR's, the Verifier can use the ADF reading to confirm the result from the VOR's.

3.2.4 The Collection of Time Related Information

The verification of parameters based on a snapshot of the system status is not always possible, because their relations may be dominated by second order effects. To verify such parameters, the IVS should have the ability to relate a series of observations and make a decision about the consistency of instrument outputs at some time (perhaps a few seconds) after an indication of trouble has been noted.

Figure 10 illustrates the condition when the EGT indication is not consistent with that of the EPR. This situation may be caused by an engine flame-out (EGT low) while the EPR-reading fails to reflect the situation. Or the situation may be caused by a defective EGT sensor that triggered the flame-out report. One possible way out of this difficulty is to make a delayed observation on N2's rpm reading since N2 will change slowly because of the rotation momentum of the engine turbine. In this example, N2 drops after a time delay, finally confirming the flame-out situation and consequently establishing the consistency of the EGT indication.
3.2.5 Diagnosis by Experimentation

After the system has done everything it can to verify the instrument outputs without modifying the aircraft status and is still uncertain about the result, there is one more step it can try. It can initiate an experiment on the aircraft control and watch for its response. Such an experiment should be sanctioned by the pilot, or should be performed with the cooperation of the pilot. This approach has been arranged as a final backup step because the pilot's attention must be diverted, which can add to his workload.

Figure 10. A delayed observation of N2 to confirm an engine failure.
Figure 11. Instrument verification by experimentation.
Figure 11 shows an interactive sequence in which the IVS asks the pilot to increase the throttle to check the Fuel Flow (FF) indicator.

3.2.6 The Implementation of the Instrument Verification System

During the second quarter, work was started on the coding of the Instrument Verification System. This work focussed on the implementation of the section which detects inconsistencies in the estimated parameter values. The ability to perform this function depends on knowledge of the aircraft, its systems and its instruments. In addition, the determination of consistency depends on the ability to propagate tolerance ranges through a computation. The parts of the knowledge base which have been implemented are discussed in the following.

3.2.6.1 Knowledge about Flight Parameters

This is the general knowledge about flight-control parameters that one may learn from flight textbooks. For example, to find the aircraft location, the following strategies, which are found in the DC-10 manuals, may be applied [10,11]:

1. Read LATITUDE and LONGITUDE from the Inertial Navigation System.
2. Compute location from 2 VOR-readings.
3. Compute location from VOR- and DME-readings.
4. Compute location from 2 ADF-readings.
5. Estimate by "Dead-reckoning".
This knowledge is organized in a LISP property-list under the name "LOCATION" and the property "TO-FIND":

```
(LOCATION ((READ-INSTRUMENT INS/LOC)
  (LOC-FROM-VOR1+VOR2)
  (LOC-FROM-VOR1+DME1)
  (LOC-FROM-ADF1+ADF2)
  (LOC-FROM-DEAD-RECKONING)
)
'TO-FIND)
```

With this representation, knowledge that relates each flight parameter to a unique set of instruments is modularly stored into the knowledge base.

### 3.2.6.2 Knowledge about Flight Instruments

This is the kind of knowledge found in a flight instrument manual:

VOR
- **READING-RANGE:** 0 - 360 degrees
- **ACCURACY:** ± 1 degree
- **TO-OPERATE:** power-on RMI
  - power-on VOR-RECEIVER
  - function-select/RMI VOR
  - freq-select/VOR-RECEIVER
    - (dialed frequency on VOR-receiver)

Again, this knowledge is stored into the knowledge base in the form of a LISP property-list:

```
('VOR (0 : 360 degrees) 'READING-RANGE)
('VOR (+ 1 degree) 'ACCURACY)
('VOR (AND (POWER RMI ON)
  (POWER VOR-RECEIVER ON)
  (FUNCTION-SELECT RMI VOR)
  (FREQUENCY-SELECT VOR-RECEIVER
    ?STATION-FREQUENCY))
)
'TO-OPERATE)
```
3.2.6.3 Knowledge about Ground Navigation Stations

This is the kind of information that is found in the Aerodrome facility directory. Such knowledge is represented in LISP as a list of N-tuples:

```
VOR-STATIONS
((NAME . CMI)
 (LOCATION . ((N 40 02 04) (W 88 16 34)))
 (FREQUENCY . 110.0)
 (CHANNEL . 37))
...
```

---

**Figure 12.** Propagation of tolerance ranges.
3.2.6.4 Tolerance Propagation

It is often difficult to determine, based on single-valued results, whether two estimations of a parameter are consistent with each other because instruments and sensors are subject to built-in errors. Such errors should be accounted for during computation by propagating them toward the final results. Figure 12 illustrates the format for a data representation which specifies a normal value within a certain tolerance range. A set of tolerance propagating operators, called the TP-operators, which combine and propagate data with its tolerance attachment, were implemented.

The LISP forms to represent values with tolerance attachments are as follows:

<table>
<thead>
<tr>
<th>LISP-form</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X (XL XU))</td>
<td>X with lower-limit XL and upper-limit XU</td>
</tr>
<tr>
<td>(X (% P))</td>
<td>= (X (X-X<em>P% X+X</em>P%))</td>
</tr>
<tr>
<td>(X (+ DX))</td>
<td>= (X (X-DX X+DX))</td>
</tr>
<tr>
<td>X</td>
<td>(X (X X)) or (X (% 0))</td>
</tr>
<tr>
<td>(X (? XU))</td>
<td>X with unknown lower-limit</td>
</tr>
<tr>
<td>(X (XL ?))</td>
<td>X with unknown upper-limit</td>
</tr>
<tr>
<td>(X (? ?,))</td>
<td>X with unknown upper and lower limits</td>
</tr>
<tr>
<td>(X (% ?))</td>
<td>= (X (? ?))</td>
</tr>
<tr>
<td>?</td>
<td>= (? (? ?)) indefinite data</td>
</tr>
</tbody>
</table>

The above forms are designed to provide a convenient user interface. The TP-operators logically propagate the indefinite "?", and the result of evaluating each formula for a given parameter is a set of values with their
respective tolerances. Two results are then said to be consistent if there is a common value in the tolerance ranges of both results.

Figure 13 illustrates a scheme for determining the aircraft location by using different combinations of instrument systems. The navigation systems available in this example are assumed to be VOR1, VOR2, DME1, ADF1, ADF2. The tolerances for the VOR-readings are within 1 degree of accuracy, for the ADF-readings, 3 degrees, and for the DME-readings approximately 1% of the total indicated distance. At least three estimations of the current location can be computed from these instrument readings and their corresponding tolerances as is illustrated in Figures 14-16. The consistency of these navigation instruments is decided from the tolerance-ranges of results.

3.3 The Script Based Monitor

The function of the computer monitor is to continuously observe the flight environment and evaluate the situation for possible errors that would threaten the safety of the flight. The use of a computer monitor can significantly reduce the pilot workload by relieving the pilot of the tedious task of monitoring numerous instrument readings and mentally integrating the raw data to determine the overall system performance. The computer monitor would be especially useful during periods of high workload when the pilot's attention is at a premium. Another advantage of the computer monitor is that the monitor would not be affected by typical human failings such as boredom, fatigue, or fixation.
Figure 13. Three schemes for determining the aircraft location.

Figure 14. Aircraft location computed from VOR1 and VOR2.
Figure 15. Aircraft location computed from ADF1 and ADF2.
Figure 16. Aircraft location computed from VOR1 and DME1.

Result:
$1 \times 1.5 \text{ N.M. SQ}$

N

VOR1

$16.7^\circ \pm 1^\circ$

CMI

( N 40 02 04)
(W 88 16 36)

DME1

77.4 N.M. ± 1%
We have completed a study of the desired attributes of a computer monitor [1], and have determined them to be the following:

1. Verification of instrument readings.
2. Detection of instrument reading deviations.
3. Diagnosis of a common cause for related deviations.
4. Report of appropriate deviations and/or diagnosis to the pilot.
5. Automatic checklist presentation.
6. Generation of a history of deviations and significant events for later analysis.

The monitor's awareness of the flight environment requires the measurement of all significant aircraft parameters via sensors. The instrument readings are verified for logical consistency. The verified instrument readings are compared against a context sensitive model for agreement. A diagnosis is made based upon the deviations to determine if there is a probable common cause. Any diagnosis and its supporting evidence are reported to the pilot when appropriate and are recorded for later analysis.

The crux of the detection of deviations lies in the determination of the normal state of the aircraft, which requires an awareness of the flight context. There seem to be two types of context in the flight domain. One type of context is a phase-based context which is of a temporal nature. The phase-based context reflects the highly regular characteristics of the flight as can be seen in Figure 17. For example, it can be expected that the takeoff roll phase follows the taxiing phase and precedes the climb phase. The phase-based context is reflected in the crew checklist where different modes are specified for different phases of flight. The other type of context is a condition-based context. Often the normal state of the aircraft or the normal
Figure 17. A three level profile of a flight.
procedure for the flight will change with the occurrence of an event that may be either internal or external to the airplane. For example, if there has been a generator malfunction, then the normal state for the bus-tie relay should be open. Also when flying under gusty wind conditions, it is normal to increase the airspeed when maneuvering close to the ground.

It is important for the monitor to be aware of the phase-based context since this context defines a normal flight. We have designed a data structure called "script" to hold the information that describes the various phases of a flight [12]. A script is defined below:

```
<script>::= ( <name>
  <entry condition>
  <next scripts>
  <normal states>
  <ranked concerns>
  <checklist>
  <procedure-scripts> )

<name>::= the script name.

<entry condition>::= key system parameters that signal this context.

<next scripts>::= possible next contexts.

<normal states>::= the normal state of the aircraft. This is the context sensitive model.

<ranked concerns>::= ordered system parameters used for conflict and priority resolution.

<checklist>::= explicit checklist if available.

<procedure-scripts>::=<script>*, also the sequence of contexts that make up this script. The <script>* also decomposes into actions.
```
The script is a tree structure. At the top, the context description is vague. The context description becomes more precise as we progress down the tree. The script is also a recursive data structure. The <procedure-scripts> nonterminal of a given script may be composed of a sequence of scripts, which make up a more precise description of the present script. Thus, each context, represented by a script structure, may be decomposed into a sequence of more finely detailed contexts also represented by scripts. The top level script is shown in Figure 18 and contains only the pointers to its three descendant scripts. One of its descendants, the takeoff phase, looks like Figure 19. The Takeoff-s script is necessarily vague since the takeoff phase covers a multitude of contexts.

```
(FLIGHT-S NIL
   NIL
   NIL
   NIL
   NIL
   (TAKEOFF-S CRUISE-S LANDING-S))
```

Figure 18. The Top Level Script

```
(TAKEOFF-S (FUEL = MAX
              ALT = 0 -> LOW)
           (CRUISE-S)
           (FUEL = MAX
              THRUST = 0 -> MAX
              ALT = 0 -> LOW
              GEAR = DOWN)
           NIL
           NIL
           (START-ENGINE-S TAXIING-S
            TAKEOFF-ROLL-S CLIMB-1-S
            CLIMB-2-S))
```

Figure 19. A Second Level Script
The twelve third level scripts shown in Figure 17 have been implemented. Only the <entry condition>, <next scripts> and <normal state> fields of these scripts have been implemented in detail. The <checklist> and <ranked concerns> fields have been implemented for some of the scripts. The preliminary monitor tracks the context transitions and the script transitions, and detects errors in a context sensitive fashion [3,13,14].

The condition-based context is to be implemented through event-action rules. These rules will implement the irregular, though frequently occurring, events that either alter the normal state of the aircraft or the normal procedure for the flight. Examples of such rules are the following:

1. generator malfunction → bus tie relay open.
2. gusty wind condition → increased airspeed when maneuvering close to the ground.

The emergency procedures for the aircraft can be implemented through these event-action rules. A data base of the emergency procedures and other flight knowledge should prove to be very useful.

The preliminary monitor system consists of the modules shown in Figure 20. The system presently consists of four programs running concurrently under time-sharing. The system environment is described by a state vector that describes the aircraft's internal systems and its trajectory in 3-D space. The state vector manager updates the vector and sends out the state vector to the monitor and the display program. The display program outputs the pertinent state vector entries on a terminal screen so that the human observer can determine the present flight environment. The monitor periodically scans the state vector and checks for context transitions and system errors as defined by the scripts in the script data base. The fault insertion program causes
Figure 20. Block diagram of the run time system.
faults in the system by altering the state vector. Presently, the flight environment is updated by the fault insertion program. An aircraft simulator has been completed, but it is not linked to the system yet. All communications to the state vector manager are routed through the IPCF facility in the TCPS-10 operating system for the CSL PDP-10 computer.

In the next quarter, we plan to implement the event-action rules for the condition-based contexts. The sufficiency of the script structure will be investigated. The cause-effect structures necessary for a more general computer monitor will also be investigated.

3.4 The Aircraft Simulator

In order to evaluate the verification and monitoring strategies being developed for the SECURE system, it is necessary to have a test bed in which to try these ideas. The aircraft simulator fulfills this function. The simulator was designed to allow for the following:

1. It must simulate relevant aircraft systems but avoid excessive detail.
2. It must be easy to implement.
3. It must be easy to modify.
4. It must be modularized.
5. It must allow faults to be introduced into the aircraft.

Since the simulator is intended to act as a test bed for the SECURE system, it should be capable of simulating all the aircraft systems with which the SECURE system will have to interact, but at the same time it should avoid superfluous detail. Thus a simple aerodynamic model is appropriate but a
simulation of a galley electrical system is not. As the SECURE system evolves, its test bed will have to evolve also. Thus ease of modification and modularity are important. Since the main purpose of the SECURE system is to aid the pilot when the aircraft is not functioning properly, the simulator should be able to simulate faults in the aircraft systems.

Figure 21 shows the basic structure of the simulator. The pilot inputs to the simulator via the standard controls, elevator, aileron etc. The state variables reflect the simulator's view of the current state of the aircraft, that is, its position, speed, fuel quantity, etc. The Dynamics procedures are used to update the state variables. Sensors provide a mechanism for SECURE and the pilot to view the state of the aircraft. It should be noted that the sensors and the state variables may disagree if a sensor fault has been introduced into the simulator. The experimenter has two inputs to the simulator. He may cause any aircraft system to appear to have failed. He may also modify any variable in the simulator.

The simulator is roughly that of a three engine commercial jet. The major activities of the simulator thus far center around maintaining the aircraft's position, speed and attitude via a simple aerodynamic model. The aircraft systems which are currently being simulated are summarized below:

1. Flight system
   controls - aileron, elevator, flaps, speed brake
   attitude - pitch, bank
   speed - air speed, rate of climb
   altitude - radio altimeter, barometric altimeter

2. Navigation system
   inertial nav. - latitude, longitude, heading
   radio nav. - VOR, DME

3. Engine system
   thrust
   fuel - fuel flow, fuel quantity
Figure 21. Diagram of the aircraft simulator.
4. Miscellaneous

4.1 Landing gear
time

Faults are introduced into the simulator by breaking the update path to certain variables in the simulator. These variables will remain at their current value unless changed by the action of the experimenter. For example, a flameout of the number one engine is introduced by failing the variable VTHRUST1 and then setting VTHRUST1 to zero. Similarly, the failure of the barometric altimeter is accomplished by failing the sensor variable SBARALT and then setting SBARALT to an erroneous value.

During the next quarter, we plan to link the simulator to the other programs in the intelligent monitoring system. In addition, the simulator will be expanded to include a more detailed simulation of some of the internal systems of the aircraft.
4 DISCUSSION AND FUTURE PLANS

The first half year of this project has seen considerable progress in the development of the conceptual framework of the SECURE system. Of the six tasks specified in the contract, the first two are nearly completed. Several interviews were conducted with flight crew members to determine the tasks which the computer will perform and the results of this study are being written in a separate report. In addition, most of the information to be included in the knowledge base has been established and some of it has been implemented as parts of the various software subsystems. Task 3, to design a planning system has begun and will be continued during the second half of the year. Finally, Task 4 has been started with the coding of parts of the instrument verification system, the script based monitor and an aircraft simulator which communicates with the IVS and the SBM. Work will continue during the next six months with the development of the concepts of the SECURE system and the implementation of these concepts. Plans are currently being formulated for the system demonstration which will be given toward the end of the contracting period.
REFERENCES


2. Rieger, C. The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension, Technical Report 373, University of Maryland, College Park, Maryland, May, 1975


13. Cullingford, R. E., "Inference in Story Understanding", Proc. 5th Intl' Joint Conf. on Artificial Intelligence, MIT, Cambridge, Ma., August 1977.