USING INFORMATION MANAGEMENT TO INTEGRATE SMART VEHICLE SUBSYSTEMS

Mark Silbert
Naval Air Warfare Center, Aircraft Division
Code 4.5.5.1.5.
Warminster, PA 18974
USA

INTRODUCTION

There are many subsystems on-board the modern military aircraft to provide enhanced capability and to enable the pilot to better perform his mission. Unfortunately, most of these subsystems each require some management by the pilot in the form of monitoring gauges, displays or indicators. As a result, pilot workload often increases since the pilot must monitor the health of his vehicle in addition to performing his mission. But the pilot should be focused on his mission, not on the aircraft itself. There are simply too many potential distractions from vehicle status indicators and gauges. Vehicle management systems need to be developed to provide more autonomy in the management of aircraft subsystems. Furthermore, when a subsystem does fail, it is left to the pilot to determine both how the failure will affect other subsystems and more importantly, how the failure will affect his ability to carry out his mission. This paper addresses the issue of how to reduce the pilot's management of aircraft subsystems by developing an integrated vehicle manager that relies on various smart subsystems technologies.

BACKGROUND

There is technology work on-going to develop various smart vehicle subsystems. These smart subsystems include smart structures, smart actuators, and smart skins. Smart structures technology is developing ways to determine stress and fatigue in the structural components of the aircraft so that these parts can be replaced before they fail. Smart actuators use embedded microprocessors and hydraulics so that they are controlled by electrical commands instead of fluids. Smart skins technology is developing embedded sensors and antennas within the skins of the aircraft.

Each of these smart subsystems technologies, however, is being developed independently of the others. Although it is intended that future aircraft will be outfitted with a variety of these smart subsystems, very little work is being done to determine how to design a vehicle manager that will coordinate and exploit these smart subsystems. Without such a manager, responsibility will fall to the pilot to manage these subsystems. An effort has begun to design a smart vehicle manager that will not only automatically manage these smart subsystems, relieving the pilot of these duties, but will also result in a more robust and versatile aircraft. This paper will describe an effort to develop this vehicle manager to manage these smart subsystems, allowing the pilot to remain focused on his mission.

PROBLEM

As mentioned, there are many smart subsystems under development but little regard is given to the possible interaction
that can occur among these subsystems on the aircraft. As a result, these smart subsystems may provide additional functionality to the pilot, but a more confusing array of failures and interactions are possible. In addition, when a failure occurs, the pilot may be alerted to the problem, but the impact on the mission is left for the pilot to determine. It is far more desirable to have the system report how the failure will affect the capability of the aircraft than to provide the failure data itself. Thus, there is a need to develop a smart vehicle manager that can exploit the capabilities of the smart subsystems to correct failures and problems, and then report to the pilot how the failure was fixed, and how the fix will affect the capability of the aircraft.

**APPROACH**

It is natural to think of these smart subsystems organized as intelligent agents communicating and interacting through a common database. By communicating through a common database, the agents share status information and cooperate in determining the appropriate fix for the problems that occur. This formalization yields a distributed problem solving approach. Blackboard architectures are useful organizations to accomplish distributed problem solving. A generic blackboard architecture is shown in figure 1.

As shown in the figure, a blackboard architecture has a number of knowledge sources and a blackboard. Each knowledge source knows how to solve part of the overall problem and tackles any subproblems that require its expertise. The blackboard is a common data area that allows the knowledge sources to share information and to communicate. It is straightforward to map the blackboard architecture onto the aircraft management problem. Each smart subsystem is a knowledge source and the common database is the blackboard. The overall goal for the subsystems is to maximize aircraft performance. The subproblems are the various subsystem failures that can occur.

In order to demonstrate the smart vehicle manager concept, three smart subsystems were selected: smart structures, smart engine accessories, and ground-collision avoidance. A software simulation of these three subsystems was developed and integrated into a man-in-the-loop simulator. It was intended to have pilots fly the simulator with and without the smart vehicle manager. Comparisons would then be made to determine the benefit of this manager.

![Figure 1. A generic blackboard architecture.](image-url)

In the following sections, each of the three smart subsystems selected will be described. The description will include the actions of the smart subsystem with and without the vehicle manager.

**Smart Structures**

There is much technology study in the development of smart structures. The primary intent is to determine stress and fatigue of aircraft parts, so that they can be replaced before they fail. The approach is to mount numerous strain/stress gauges throughout the aircraft frame and measure the onset and degree of fractures. Engineers involved in smart structures see this technology primarily as a way to support the maintenance personnel, not the pilot. These engineers envision their
sensors recording the structural integrity of the aircraft during the mission, and then having maintenance personnel review the tapes once the aircraft is on the ground. Smart structure engineers expect to alert the pilot only if the crack is catastrophic and the pilot must eject.

Without a vehicle manager, very little information is provided to the pilot. If minor structural damage occurs, the smart structures will detect and record this damage but leave the pilot unaware. But this nondisclosure could have catastrophic results. For example, minor damage could become severe if the pilot pulls excessive G forces or sustains high speeds. It would be more desirable to maintain the G and speed limits of the aircraft and alert the pilot when they need to be reduced. A smart vehicle manager would keep the pilot aware of these maximum G and speed limits and prevent the pilot from exceeding them.

**Smart Engine Accessories**

There are various power systems on-board the aircraft that are closely coupled. When a failure occurs in one of these systems, the interactions could lead to a confusing array of warnings and cautions to the pilot. During the confusion, the pilot could easily fail to execute the right sequence of controls to alleviate the problem. Furthermore, this could be quite stressful if the failure occurs at a critical time of the mission.

Figure 2 shows a simplified diagram of the ECS/power system in a typical aircraft. From the figure, it can be seen that the environment control system (ECS) for the cockpit works as an air bleed off the engine. In addition, the aircraft mounted auxiliary device (AMAD) connects to the shaft of the engine to power the generator and the hydraulic pump. Fuel is used to cool the AMAD.

With smart engine accessories, the subsystem automatically checks for hot gas leaks and fires in or around the engine. For example, suppose a leak occurs in the right ECS. This hot gas could eventually ignite the fuel in the right feeder tank. The hot gas will eventually trigger both the right ECS warning light and the right fire warning light. A pilot acknowledges the fire warning by pressing the fire warning switch. When this switch is pressed, the aircraft shuts off all fuel to the right engine and the engine stops running. The pilot may choose to restart the engine by first supplying fuel to the right engine. As shown in figure 2, the pilot can supply fuel to the engine by either restarting the right fuel pump or by enabling a crossfeed. While it may be safer to crossfeed fuel to the engine when the possible fire conditions occur, the aircraft should not be flown this way for prolonged periods of time. As shown in the figure, fuel is used to cool the AMAD, but when a crossfeed is enabled, the AMAD cooling is bypassed. After awhile, the AMAD will become very hot and may cause a fire as well.

![Figure 2. A typical aircraft ECS/power system.](image)

The smart engine accessories subsystem will warn the pilot of a possible fire in the engine bay, but it will not take much action on its own to prevent a possible fire from starting. On the other hand, a vehicle manager would monitor the status of these power systems and make reasonable responses to avoid a possible fire. For example, suppose the pilot continues to fly by crossfeeding fuel to
the right engine. The vehicle manager will sense the AMAD getting hot and instead of warning the pilot about the hot AMAD, the manager will disengage the clutch connecting the AMAD to the engine. The only major effect will be possibly reduced responsiveness of the stick at slow speeds due to the reduced hydraulic pressure. The smart vehicle manager will alert the pilot about the action taken, tell why it was taken, and convey the resulting effect on aircraft performance. So instead of a warning light indicating a hot AMAD, the pilot will get a message like:

AMAD disengaged due to hot AMAD
Stick may be sluggish below 300 KTS

Smart Ground Collision Avoidance

There are many engineers developing ground collision avoidance systems (GCAS). The intent of these systems is to prevent the pilot from flying into the ground below him or a mountain in front of him. Many pilots are uncomfortable with these systems. The pilots argue that the aircraft is not aware of the maneuvers the pilot intends to do or the amount of flying expertise he has. While this is generally true, a smart vehicle manager would allow the pilot to customize the GCAS, providing more control over the system’s behavior. Furthermore, since the vehicle manager maintains the G and speed limits of the aircraft, the GCAS would only take control if any avoiding maneuver would exceed these limits. Through the vehicle manager, the pilot can customize the GCAS to specify how close to the ground the pilot intends to fly and how much look-ahead the GCAS should consider. When the GCAS takes control of the stick, the vehicle manager would display a big arrow indicating the direction of the pull up. By displaying the arrow, the pilot is not only alerted to the fact that the GCAS took control, but the direction the pilot should take when control is returned to him.

Discussion

Due to severe time and budget constraints, only one fighter pilot flew the simulation and some of the GCAS functionality was not present. Specifically, there was no capability to customize the GCAS. In our very limited study, there were several lessons learned. First, the tolerance threshold determining when the GCAS should take control was much too tight. The pilot complained that the GCAS was much too sensitive. This stressed the need for the pilot to be able to customize the GCAS parameters. Unfortunately, this biased the pilot’s feelings about the GCAS system and actually raised his frustration level during those missions with the vehicle manager. The benefit of the vehicle manager for controlling the power systems failures went somewhat unnoticed. For example, the pilot apparently never resorts to crossfeeding fuel. His feelings were that if the engine is on, the fuel pump might as well be on too. Thus, the pilot never

Prototypes of these three smart subsystems were implemented in C and then integrated into a man-in-the-loop simulation. A fighter pilot flew the simulated aircraft through several runs of a specified JDAM mission, with and without the vehicle manager. During each run, the pilot was told whether the vehicle manager was present. Any failures that might occur during the flight were not disclosed to the pilot beforehand. During some of the runs, the simulation would simulate structural and/or power system failures. The GCAS was enabled only during those runs where the vehicle manager was present. Both quantitative and qualitative data were gathered for each run. Examples of quantitative results include the number of times the pilot flew into the ground, and the amount of times the pilot deviated from the intended track. Qualitative data included asking the pilot the level of frustration and difficulty he experienced during the run, on a 10-scale.

RESULTS

Due to severe time and budget constraints, only one fighter pilot flew the simulation and some of the GCAS functionality was not present. Specifically, there was no capability to customize the GCAS. In our very limited study, there were several lessons learned. First, the tolerance threshold determining when the GCAS should take control was much too tight. The pilot complained that the GCAS was much too sensitive. This stressed the need for the pilot to be able to customize the GCAS parameters. Unfortunately, this biased the pilot’s feelings about the GCAS system and actually raised his frustration level during those missions with the vehicle manager. The benefit of the vehicle manager for controlling the power systems failures went somewhat unnoticed. For example, the pilot apparently never resorts to crossfeeding fuel. His feelings were that if the engine is on, the fuel pump might as well be on too. Thus, the pilot never
experienced the hot AMAD problem as caused by prolonged crossfeeding. The smart structures, on the other hand, was a huge success. During those simulated runs where the pilot had structural damage and no vehicle manager, the pilot ended up ripping a wing off the plane! However, when the vehicle manager was enabled, the pilot remained in control subsequent to the damage and flew the mission safely. In fact, the pilot did not notice that his stick was being restricted by the vehicle manager.

Overall, the pilot had mixed feelings about the smart vehicle manager. As previously mentioned, the pilot remarked that his frustration level was higher during the runs with the vehicle manager. Part of the frustration was due to the fact that the GCAS couldn’t be customized. The other part was that the pilot got a bit unnerved when the vehicle manager told him it was taking care of problems for him. The pilot had a bit of distrust in the system so he found himself “double-checking” what the manager had done. The question naturally arises of whether the pilot would ever gain trust in the vehicle manager.

CONCLUSION

Although our results did not indicate a urgent need for a vehicle manager, it showed there is much promise because of the success of the smart structures component. This initial work helped show the concept of exploiting the “smarts” in the various smart subsystems to design a more robust aircraft. It is clearly necessary, however, to more fully develop the vehicle manager and test it with many more pilots. Work has already begun to enhance the GCAS and to integrate other smart subsystems into our vehicle manager information management framework.