NASA-CR-205043

An Agent-Based Cockpit Task Management System

NASA grant NAG 2-875

Final Report

Ken Funk
Principal Investigator

JUL 2 2 1997
C. A.S. I.

Department of Industrial and Manufacturing Engineering
118 Covell Hall
Oregon State University
Corvallis, OR 97331-2407

30 June 1997

An Agent-Based Cockpit Task Management System

The objectives of this research were to develop and evaluate an agent-based program to facilitate Cockpit Task Management (CTM) in commercial transport aircraft. During the course of this research we refined the concept of CTM and renamed it Agenda Management (AMgt). We developed an agent-based program called the AgendaManager (AMgr) and evaluated it in a part-task simulator study using airline pilots. Results from the study indicate that the AMgr was in fact effective in facilitating AMgt.

As a final report on this project we present copies of the following proceedings and journal articles published or submitted for review during the grant period. The references and the papers themselves appear in the order in which the work was done and not in order of the date of publication.

Chou, C.D., Madhavan, D. and Funk, K. (1996). Studies of cockpit task management errors, *International Journal of Aviation Psychology*, 6(4), 307-320.

Kim, J.N. & Funk, K. (1995). An aid to facilitate cockpit task management performance. In *Proceedings of the Eighth International Symposium on Aviation Psychology* (pp. 196A-196F). Columbus, OH: The Ohio State University Department of Aviation.

Funk, K. and Kim, J.N. (1995). Agent-based aids to facilitate cockpit task management. In *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics* (pp. 1521-1526). Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Funk, K., & McCoy, B. (1996). A functional model of flightdeck agenda management. In *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting* (pp. 254-258). Santa Monica, CA: Human Factors and Ergonomics Society.

Funk, J. & Braune, R., (in press). Expanding the functionality of existing airframe systems monitors: the AgendaManager. In *Proceedings of the Ninth International Symposium on Aviation Psychology*. Columbus, OH: The Ohio State University Department of Aerospace Engineering, Applied Mechanics, and Aviation.

Cha, W.C. & Funk, K. (in press). Recognizing pilot goals to facilitate agenda management. In *Proceedings of the Ninth International Symposium on Aviation Psychology*. Columbus, OH: The Ohio State University Department of Aerospace Engineering, Applied Mechanics, and Aviation.

Cha, W.C. & Funk, K. (in review). Communicating pilot goals to an intelligent cockpit aiding system, *Engineering Design and Automation*.

Funk, K., Cha, W.C., Wilson, R., Zaspel, J, & Braune, R. (in press). Development and evaluation of an aid to facilitate agenda management. In *Proceedings of the 1997 IEEE Conference on Systems, Man, and Cybernetics*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Funk, K., Cha, W.C., Wilson, R., Zaspel, J, Braune, R., & Kim, J.N. (1997). *Agenda Management: Understanding and Facilitating the Management of Flightdeck Activities* (world wide web site). Corvallis, OR: Oregon State University Department of Industrial and Manufacturing Engineering, http://www.engr.orst.edu/~funkk/AMgt/AMgt.html.

The following PhD dissertations were completed under this grant, but copies are not included in this report.

Kim, J.N. (1994). *An Agent-Based Cockpit Task Management System: a Task-Oriented Pilot-Vehicle Interface*, unpublished doctoral dissertation. Corvallis, OR: Oregon State University Department of Industrial and Manufacturing Engineering.

Cha, W.C. (1996). *Communicating Pilot Goals to an Intelligent Cockpit Aiding System*, unpublished doctoral dissertation. Corvallis, OR: Oregon State University Department of Industrial and Manufacturing Engineering.

The following MS theses were completed under this grant or are in final preparation, but copies are not included in this report.

Wilson, R.B. (1997). *Development and Comparison of Two Alternate Display Formats for an AgendaManager Interface*, unpublished master's thesis. Corvallis, OR: Oregon State University Department of Industrial and Manufacturing Engineering.

Zaspel, J. (in preparation). *A Fuzzy Logic Based System to Assess Flight Function Performance* (tentative title), unpublished master's thesis. Corvallis, OR: Oregon State University Department of Industrial and Manufacturing Engineering.

Copies of the dissertations and theses are available from the principal investigator.

FORMAL PAPERS

Studies of Cockpit Task Management Errors

Chung-Di Chou

Distributed Systems Department
MBNA America

Das Madhavan

Performance Engineering Department
Household Credit Services Inc., Salinas, California

Ken Funk

Department of Industrial and Manufacturing Engineering
Oregon State University, Corvallis

Cockpit task management (CTM) is the management level activity pilots perform as they initiate, monitor, prioritize, and terminate cockpit tasks. To better understand the nature and significance of this process, we conducted 3 empirical studies: a review of National Transportation Safety Board aircraft accident reports, a review of Aviation Safety Reporting System aircraft incident reports, and a simulator experiment. In the accident report study, we determined that CTM errors occurred in 76 (23%) of the 324 accidents we reviewed. We found CTM errors in 231 (49%) of the 470 incident reports we reviewed. In the simulator study, we found that CTM performance was inversely related to workload. We conclude that CTM is significant to flight safety and recommend that this realization be reflected in pilot training, in cockpit procedures, and in research to develop pilot aiding systems.

Requests for reprints should be sent to Ken Funk, Industrial and Manufacturing Engineering, Oregon State University, 118 Covell Hall, Corvallis, OR 97331–2407.

Flight crews not only have to perform individual tasks to accomplish missions; they must manage tasks as well. They must make sure that tasks are started and stopped at the right times and that undue attention to lower priority tasks does not prevent the correct and timely completion of higher priority tasks. Just as failure to perform tasks correctly can lead to accidents, failure to manage tasks correctly can have catastrophic consequences as well.

This article describes three related studies that have helped us to understand the nature and significance of task management in commercial flight operations: a study of aircraft accidents, a study of aircraft incidents, and a study of task management behavior in the laboratory.

BACKGROUND

We define a *task* as a process performed (at least partly by a human) to achieve a goal, such as to fly to a waypoint, descend to a desired altitude, obtain a clearance from air traffic control, or restart an engine. Most human factors and engineering psychology researchers have focused on the task as the unit of human behavior, and many theories of task performance and errors have emerged.

A less prominent line of research has addressed the higher level activity of managing multiple, concurrent tasks. For example, Johannsen and Rouse (1979) introduced the notion of a time-sharing computer system as a metaphor for human multiple task performance but did not address in any detail the nature of the executive task, that task responsible for managing other tasks. In her studies of workload, Hart (1989) found that participants attempted to maintain a relatively constant level of workload by means of a form of task management: shedding or assuming tasks as workload increased or decreased. Moray and his colleagues (1991) proposed scheduling theory as a normative model for how operators manage multiple tasks and found that unaided human participants adopted suboptimal scheduling strategies. In a simulator study, Raby and Wickens (1994) investigated the effect of workload on task management, finding that as workload increased, participants adjusted task performance strategies.

Our research is based on a theory developed by Funk (1991) to describe task management behavior in the cockpit domain. According to this theory, cockpit task management (CTM) consists of the following functions:

1. *Task initiation*: The initiation of tasks when appropriate conditions exist.
2. *Task monitoring*: The assessment of task progress and status.
3. *Task prioritization*: The assignment of priorities to tasks relative to their importance and urgency for the safe completion of the mission.
4. *Resource allocation*: The assignment of human and machine resources to tasks so that they may be completed.

5. *Task interruption*: The temporary suspension of lower priority tasks so that resources may be allocated to higher priority tasks.
6. *Task resumption*: The resumption of interrupted tasks when priorities change or resources become available.
7. *Task termination*: The termination of tasks that have been completed, that cannot be completed, or that are no longer relevant.

Objectives

The broad purpose of our research was to determine the nature and significance of CTM in flight operations and, if appropriate, make recommendations to improve it. To achieve this, the following specific objectives were formulated:

1. Develop a taxonomy of CTM errors.
2. Study CTM behavior in operational settings by means of accident and incident reports.
3. Study CTM behavior under controlled laboratory conditions.
4. Make recommendations to improve CTM behavior through training and design.

The organization of this article follows that of the objectives.

CTM Error Classification

Chou and Funk (1990) developed an initial CTM error taxonomy consisting of seven general CTM error categories corresponding to the aforementioned seven functions of CTM. Each category was further described in terms of specific error classes. Use of the initial taxonomy in preliminary analyses of accident and incident reports showed some of the error classes to be redundant and the taxonomy, as a whole, to be difficult to apply consistently.

As a result, we revised the taxonomy to include the CTM error categories shown in Table 1. To summarize, a task may be initiated or terminated too early, too late, under incorrect conditions, or for incorrect reasons; or it may not be initiated or terminated at all. Furthermore, a task may be given too high or too low a priority. This revised taxonomy served as the basis for our accident and incident studies, descriptions of which follow.

CTM Errors in Aircraft Accidents

The underlying causes of aircraft accidents usually fall into the three broad categories of mechanical factors, weather, and pilot error. However, these labels

TABLE 1
CTM Error Taxonomy

Error Categories	Possible Classifications
Task initiation	Early
	Late
	Incorrect
	Lacking
Task prioritization	Incorrect
Task termination	Early
	Late
	Incorrect
	Lacking

should not be used to mark the end of further analyses for human and other system performance errors because aircraft accidents are usually the outcomes of a number of contributing factors. In an effort to determine whether some instances of pilot error could be explained in terms of CTM, and thereby begin to understand the significance of CTM to flight safety, we reviewed a set of aircraft accident reports (Chou, 1991).

Our analysis reflects the examination of the abstracts of 324 National Transportation Safety Board (NTSB) aircraft accident reports concerning accidents occurring between 1960 and 1989. After reviewing the 324 National Technical Information Service abstracts of these reports, we removed accidents that were obviously unrelated to this study from the screening process. For example, accidents due primarily to weather and mechanical failures were removed. This elimination process left 76 accident reports for further analysis.

Following the initial screening, we selected a representative set of cases for further study, based on the following considerations. First, we chose the cases so as to include a complete set of CTM errors as listed in Table 1. Second, we chose cases involving conditions we believed we could reconstruct in a simulated environment. Based on these considerations, we settled on a set of cases including the following accidents: Eastern Flight 401, a Lockheed L1011 (NTSB, 1973); China Airlines Flight 006, a Boeing 747 (NTSB, 1986a); Piedmont Flight 467, a Boeing 737 (NTSB, 1986b); Air Florida Flight 90, a Boeing 737 (NTSB, 1982); Comair Flight 444, a PA31-310 (NTSB, 1979); and a Texasgulf Aviation flight, a Lockheed JetStar (NTSB, 1981). For each accident in this set, we carefully studied the data and conclusions of the NTSB investigators and constructed an operational task context.

Each context was a graphical representation of cockpit activities during the time leading up to the accident. It included the number and type of concurrent tasks competing for the flight crew's resources, the state of each task (pending, active,

interrupted, or terminated), and selected system state variables (e.g., aircraft altitude, speed, etc.).

For example, Figure 1 shows the task context for Eastern Flight 401, a Lockheed L1011, in the last 10 min before the accident. In this accident, the flight crew became preoccupied with a possible landing gear indicator fault and failed to notice the aircraft's gradual descent, which eventually led to the crash. The upper portion of this figure shows crew activity on four concurrent tasks in this period: aircraft control (FLYING), ATC communication (COMM), diagnosis of the landing gear indicator (DIAGNS), and inspection of crew-accessible parts of the landing gear itself (INSPCT). The lower portion of the figure shows aircraft altitude and time. This figure shows our finding that the flight crew's attention was focused on the landing gear problem to the exclusion of the flight control task.

We identified this as a CTM error and classified it as a task-prioritization-incorrect error, backing up our interpretation of the data with the conclusions of the NTSB. With the insights gained from this detailed analysis and using the data and conclusions in the accident abstracts and full reports, we identified and classified 80 CTM errors in 76 of the 324 accident reports. That is, we found that CTM errors occurred in about 23% of the accidents reviewed. These errors, summarized by category, are presented in Table 2.
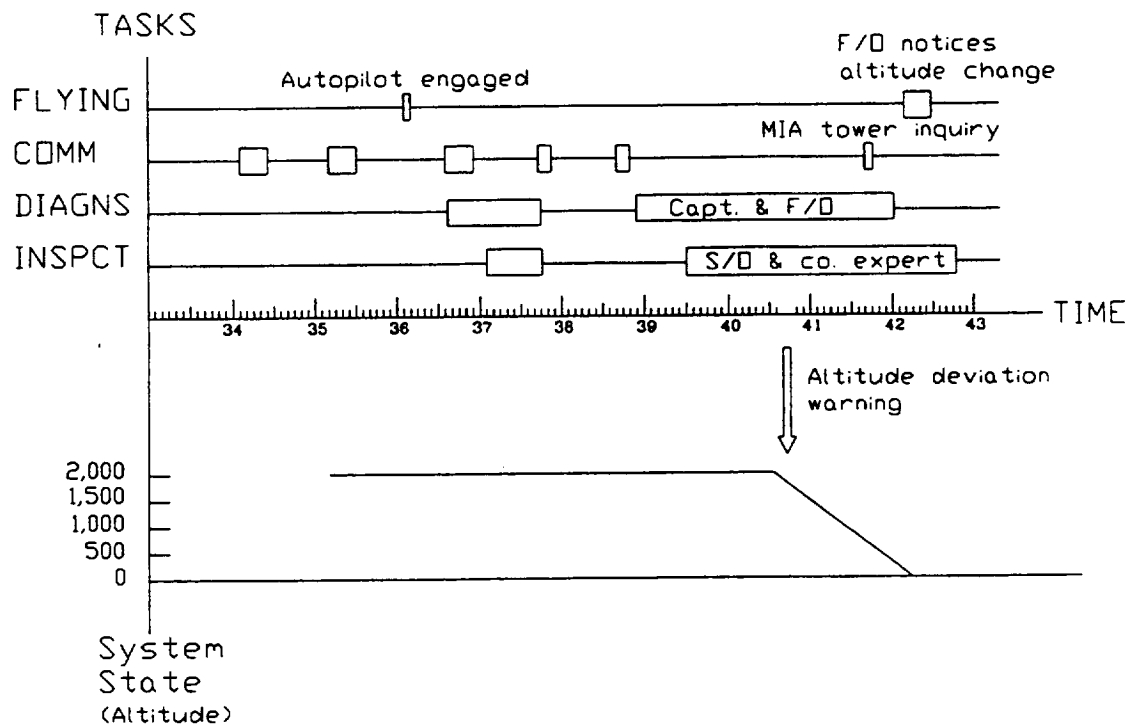

FIGURE 1   Task context for Eastern Flight 401.

TABLE 2
CTM Errors Identified and Classified in 76 (23%) of 324 NTSB Accident Reports

CTM Error	Number of Accidents	Percentage of CTM Accidents	Number of CTM Errors	Percentage of All CTM Errors
Task initiation	35	46	35	44
Task prioritization	24	32	24	30
Task termination	21	28	21	26

Note. CTM = cockpit task management; NTSB = National Transportation Safety Board. Total number of CTM errors = 80.

Although we cannot state categorically that CTM errors were the sole or even primary causes of these accidents, we do believe that they played significant roles. Had the errors been prevented, the accidents probably would not have occurred. We conclude that the moderately high incidence of CTM errors in the accidents—76 (23%) of 324 accidents—is supportive evidence that CTM is a significant factor in flight safety.

CTM Errors in Critical In-Flight Incidents

Fortunately, aircraft accidents are very rare events. Unfortunately, a set of accidents such as the one we studied might be a very biased sample of the operating environment. Therefore, inferences made from a set of accidents may have little relevance to reducing the likelihood of future accidents. For that reason, we next turned our attention to aircraft incidents (Madhavan, 1993). An incident is defined as "an occurrence other than an accident, associated with the operation of an aircraft, which affects or could affect the safety of operations" (Federal Aviation Regulations, 1994). Although incidents by definition do not involve death, serious injury, or substantial aircraft damage, it is clear in retrospect that most airline accidents were foreshadowed by clear evidence that the problems existed long before as incidents. Our specific objective in analyzing aircraft incidents was to determine the significance of CTM in flight operations more representative of normal conditions.

We used as a source of aircraft incident information NASA's Aviation Safety Reporting System (ASRS). The ASRS database consists of anonymous reports filed by pilots and air traffic controllers describing events in which accidents nearly occurred or in which flight safety was seriously compromised.

Our preliminary analysis of CTM errors focused on aircraft incident reports relating to in-flight engine emergencies (99 reports) and controlled flight toward terrain (CFTT; 205 reports). We found CTM errors in 19% and 54%, of these reports, respectively. The high incidence of CTM errors in the CFTT reports, as

well as the fact that over 49% of all airline accidents occur during approach and landing (Boeing Commercial Airplane Group, 1994, March), caused us to focus further attention on the terminal phases of flight. At our request the ASRS office furnished us with 243 additional reports pertaining to these phases.

As in most ASRS studies, we used the narrative section of the reports for our analysis. The narrative is the section of the report in which the reporter states in his or her own words what happened and why it happened.

In the narratives, we focused on activities directly related to task management only. Incidents involving crew personality differences and other sociological factors were excluded. When narratives were unclear about the specific errors committed (i.e., no categoric admission of the errors by the reporters), some inferences were made about the errors based on our knowledge of standard operating procedures as gleaned from aircraft operations manuals, accident reports, incident reports, and other aviation literature (e.g., Stewart, 1992). Key words and phrases in the narratives—such as "forgot," "omitted," "memory lapse," "oversight," etc.—enabled us to home in quickly on the error classification. As an illustration of our method, one such report (typical of the reports in this flight phase), is reproduced, in part, here (ASRS Rep. No. 144766). This excerpt is verbatim from the ASRS database except that case has been converted (ASRS reports are recorded in all uppercase letters). CTM error classifications are inserted in square brackets and are explained following the excerpt:

> Capt. was flying acft. A tornado watch had him worried and asked F/E to contact FSS to get details descending into DTW [task prioritization incorrect]. His radio interfered with COM on radio #2 which I was on with APCH. During this confusion dsnt and apch clrncs had to be repeated a few times distracting my x-chk of cpt's INS. Intercepting LOC capt went right through the LOC and saw he had 66 degs not 33, as apch calls for. I called out that and he put 33 in the window, corrected back and overshot again (APCH asked if we needed vectors back for a new apch). He said no. I said "I don't like the look of this." We had full LOC deflection and were above G/S. Capt. said "let's see how it is at 1000." At 1000' he did manage to get back on LOC and kept descending to a successful lndg [task termination lack]. Capt. had poor CRM and poor judgement. I should have said,"go missed apch," F/E should have said the same, but was still doing chklist—late [task initiation late] because of talking to FSS. It was the first time I had seen an apch so messed up! I will never allow it to happen again![1]

---

[1] Capt., cpt = captain; F/E = flight engineer; FSS = flight service station; APCH = approach control; apch = approach; dsnt = descent; clrncs = clearances; x-check = cross-check; INS = inertial navigation system; LOC = localizer; G/S = glide slope; lndg = landing; CRM = crew resource management; chklist = checklist.

The captain elected to perform a lower priority task (radio for weather update) at a critical point in the flight (final approach to land), which caused the F/E to delay his checklist. The reporter implied that the captain should have aborted the landing. From the narrative it appears that the captain's attention was allocated primarily to the tornado watch with little left for landing safely (as evidenced by his mis-setting the localizer course and continuing with the landing despite being at full localizer deflection).

From the 540 ASRS incident reports we obtained, we eliminated duplicates. We then applied the CTM error taxonomy to the remaining 470 unique reports. We found CTM errors in 231 (49%) of the 470 ASRS incident reports. The results of the analysis are presented in Table 3.

Task initiation appears to be the most significant CTM error category, accounting for 42% of the CTM errors identified. Task initiation errors included early descents, late configurations, and failures to tune navigation and communication radios. Task prioritization errors accounted for 35% of the CTM errors and included distractions by weather and traffic watches. The remaining 23% of the CTM errors were in the task termination category. These included early autopilot disengagements, altitude overshoots, and improperly continued landings under unsafe conditions.

Although task initiation appears to be the largest CTM error category, that may be somewhat misleading. The failure to start a task on time (or at all) or the decision to start a task too-early may often be explained as misprioritization. That is, excessive priority placed on one task may delay the start of a second task or cause the flight crew to start the first task before they should. Similar arguments can be made for task prioritization versus task termination. Although the initiation and termination categories are useful for understanding errors, their causes, and their consequences, task prioritization should perhaps draw our greatest attention for the development of countermeasures. We conclude that the high incidence of CTM errors in the incident reports—231 (49%) of 470 reports—is supportive evidence that CTM is a significant factor in flight safety.

TABLE 3
CTM Errors Identified and Classified in 231 (49%) of 470 ASRS Incident Reports

CTM Error	Number of Incidents	Percentage of CTM Incidents	Number of CTM Errors	Percentage of All CTM Errors
Task initiation	137	59	145	42
Task prioritization	133	58	122	35
Task termination	83	36	82	23

Note. CTM = cockpit task management; ASRS = Aviation Safety Reporting System. Total number of CTM errors = 349.

FLIGHT SIMULATOR STUDY

From our accident and incident studies, we determined that CTM is significant enough to warrant further study. However, we felt that a different approach was needed to better understand the nature of CTM behavior. Aircraft accidents are rare events, thus providing few opportunities for developing insights into error processes, which are, in any case, very difficult to reconstruct. By the same token, though ASRS incident reports can provide firsthand information on abnormal cockpit operations, they are subject to self-reporting biases and other problems. Therefore, controlled experimentation provides a useful alternative, serving to compensate for the drawbacks noted previously and to provide an opportunity for objective observations. An additional advantage of the simulation method is that it enables observation of how human operators manage tasks under normal conditions.

The main objectives of our experiment were to elicit and observe CTM errors similar to those identified in the accident and incident analyses and to identify the factors leading to such errors. Our approach was to have participant pilots fly a low-fidelity flight simulator in several flight scenarios and observe and analyze their behavior in managing and performing concurrent flight tasks.

Apparatus

Our flight simulator consisted of three networked personal computers. The system simulated a generic, two-engine commercial transport aircraft. One computer simulated aircraft dynamics using a very simple aerodynamic model and produced a simple primary flight display showing heading, altitude, airspeed, pitch, and roll. The participant controlled the simulated aircraft by means of a joystick. A second computer simulated the navigation system and presented a moving map display. The participant could use the navigation display for planning and navigating purposes and could control map scale and orientation (north up or track up) by means of mouse-activated controls. The third computer simulated aircraft subsystems, including engines and the hydraulic system, and generated a simplified engine indication and crew alerting system display. Aircraft subsystem models included failure modes that could be triggered by script files and that required participant interaction by mouse-activated controls to correct.

Participants

Twenty-four unpaid participants from Oregon State University participated in the experiment. The participants included 2 engineering faculty members, 3 undergraduate engineering students, and 19 engineering graduate students. Two of the

participants had private pilot licenses with 120 to 150 hr of flight time. The other participants had no flight experience. Sixteen participants participated in two pilot studies, and the remaining 8 participants took part in the data collection runs. The pilot studies were used for refining training procedures and flight scenarios.

Procedures

Participants received a 60-min training session prior to each experiment. This session included viewing a training videotape and running a simplified scenario. The scenarios were categorized into six different levels by the following independent variables: resource requirements, maximum number of concurrent tasks, and flight path complexity. Following concepts from multiple resource theory (Wickens, 1992) and workload index (W/INDEX; North & Riley, 1989), scenarios were created and rated according to the requirements for visual resources (to acquire needed information from simulated visual displays), manual resources (to manipulate simulated controls), and mental resources (to recognize, remember, calculate, and decide). Each scenario received an aggregate resource requirements rating (low or high). The number of concurrent tasks was defined as the maximum number of tasks requiring participant attention at any point in the scenario (three or six). Flight path complexity (easy or hard) was varied by adjusting the sharpness of turns at waypoints in the flight path.

A split-plot design (Steel & Torrie, 1980) was used for the experiment. The latter factors (number of concurrent tasks and flight path complexity) were crossed to provide four levels for whole unit factors. These four whole unit factors were then crossed with the subunit levels (resource requirements) to provide eight treatments. Given this design, eight participants were used to provide two responses for each treatment. Each participant performed two levels of the subunit factor (low and high resource requirements), and the assignment of treatments to participants was randomized to control learning effect. That is, four participants started with the high resource requirements treatment and then performed the low resource requirement treatment, whereas the other four participants performed their treatments in the reverse order.

Performance Measurement

The following performance measures were used:

1. Average response time to system faults.
2. Root-mean-square (RMS) flight path error.
3. Task prioritization score.
4. Number of tasks that were initiated late.

The response time to a system fault was defined as the time from the occurrence of the fault (such as an electrical bus fault) until a compensating response was initiated. This corresponded to task initiation. The task prioritization score was determined from paired comparisons between tasks and was used for measuring task prioritization performance. A score of +1 was assigned when a correct prioritization was made by the participant (i.e., attention was first given to the higher priority task); otherwise a −1 was assigned. Scores for the remaining tasks were set to zero. Finally, a task was said to be initiated late if the participant did not respond to the task 60 sec after it had been activated. This was used to measure task initiation performance.

Results

The analysis of variance (ANOVA) results for factors with significant effects are summarized in Table 4. We found that the resource requirements level had a significant effect on the average task response time. That is, higher resource requirements increased delays in initiating a task. However, neither combination of flight path complexity nor maximum number of concurrent tasks (alone or in combination) had a significant effect on task response time.

During the experiments, participants were warned if 60 sec passed after the occurrence of a system fault and no actions were taken. Thus, the definition of a late initiation was failure to initiate the task within 1 min following fault occurrence. The ANOVA results show that resource requirements had a significant effect on late task initiation.

Results from the ANOVA show that both resource requirements and the combination of flight path complexity and number of concurrent tasks created signifi-

TABLE 4
Summary of Experimental Results

| | Experimental Factors | | | |
| | Number of Concurrent Tasks and Flight Path Complexity (df = 3, 4) | | Resource Requirements (df = 1, 4) | |
Response Variables	F	p	F	p
Task initiation (average response time)	5.85	.060	14.65	.019
Task initiation (late task initiation)	< 6.59	> .05*	27.00	.007
Task prioritization	32.08	.003	34.13	.004
RMS flight parameter errors	1.26	.400	3.04	.156

*Not significant; exact F and p values were not recorded.

cant effects on task prioritization. Therefore, task prioritization degrades as either one of these factors increases.

We calculated the RMS of deviations in flight parameters using data obtained from whole-mission information. Heading deviations were significantly affected by the combination of flight path complexity and the number of tasks; changes in mental resource requirements were significant to the altitude deviation. None of the other RMS deviations were significantly affected by either the resource requirements or the combination of flight path complexity and the number of concurrent tasks.

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

We developed a taxonomy of CTM errors based on Funk's normative theory of CTM (Funk, 1991) and applied it in the analysis of NTSB aircraft accident reports and ASRS incident reports. We found CTM errors in 76 (23%) of the 324 accident reports analyzed and in 231 (49%) of the 470 incident reports. In a low-fidelity simulator study, we found that resource requirements (visual, manual, and mental) had a statistically significant effect on task initiation and task prioritization performance, and that the number of concurrent tasks coupled with flight path complexity had a statistically significant effect on task prioritization performance.

From our studies of aircraft accidents and incidents, we conclude that CTM is a significant factor in flight safety. And, as Raby and Wickens's (1994) results implied, our experiments confirm that increased resource requirements increase the likelihood of CTM errors—specifically, late task initiation and incorrect task prioritization errors.

We offer four recommendations. First, we recommend that pilots receive instruction concerning CTM and how to avoid CTM errors. More specifically, pilots should be made aware that in periods of high workload, when large numbers of concurrent tasks are competing for their attention, there is danger that they will not initiate important tasks promptly and that their attention will be drawn away from safety-critical tasks. Presumably, pilots can be taught to recognize these precursor conditions and to develop personal strategies to avoid CTM errors when these conditions are present. CTM instruction might most naturally fit into existing crew resource management training programs.

This recommendation is based on the assumption that our experimental environment, involving a low-fidelity simulator and (mostly) nonpilot participants is, at a very high level of abstraction, similar enough to the real commercial transport aircraft environment to warrant extrapolation. This assumption should be tested, so our second recommendation is that further studies of CTM be conducted using full-mission scenarios in high-fidelity training simulators with line pilots as par-

ticipants. The objectives should be to validate our earlier findings, to search for other factors affecting CTM performance, to identify patterns of both good and bad CTM, and to attempt to link CTM errors with human cognitive characteristics, such as short-term (working) memory limitations.

Third, we recommend that research be conducted to develop and evaluate formal cockpit procedures to facilitate CTM performance, based on findings from the studies recommended previously. Such procedures might, for example, involve memory aids and elaborated versions of the well-known pilots' prioritization maxim: "aviate—navigate—communicate."

Finally, our fourth recommendation is that research be conducted to develop and evaluate a computational aid to facilitate CTM performance: a Cockpit Task Management System (CTMS). A CTMS might, for example, perform the following functions:

1. Maintain a current model of aircraft state and current cockpit tasks.
2. Monitor task state and status.
3. Compute task priority.
4. Remind the pilots of all tasks that should be in progress.
5. Suggest that the pilots attend to tasks that do not show satisfactory progress.

We must point out, however, that for any approach to be effective, net pilot workload must not increase. If personal strategies, formal procedures, or computational aids impose additional mental demands, there must be compensatory workload reductions. Otherwise, the supposed aids may actually lead to even worse CTM performance.

ACKNOWLEDGMENT

REFERENCES

Aviation Safety Reporting System. *Aviation Safety Reporting System Database*. Moffett Field, CA: ASRS Office, NASA Ames Resaerch Center.

Boeing Commercial Airplane Group. (1994, March). *Statistical summary of commercial jet aircraft accidents, worldwide operations, 1959–1993*. Seattle, WA: Airplane Safety Engineering, Boeing Commercial Airplane Group.

Chou, C. D. (1991). *Cockpit task management errors: A design issue for intelligent pilot–vehicle interfaces*. Unpublished doctoral dissertation, Oregon State University.

Chou, C. D., & Funk, K. (1990). Management of multiple tasks: Cockpit task management errors. *Proceedings of the 1990 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 470–474). Piscataway, NJ: The Institute of Electrical and Electronics Engineers.

Federal Aviation Regulations, 49 C.F.R. § 830.2 (1994).

Funk, K. (1991). Cockpit task management: Preliminary definitions, normative theory, error taxonomy, and design recommendations. *The International Journal of Aviation Psychology, 1,* 271–285.

Hart, S. G. (1989). Crew workload management strategies: A critical factor in system performance. In R. S. Jensen (Ed.), *Proceedings of the Fifth International Symposium on Aviation Psychology* (pp. 22–27). Columbus: The Ohio State University, Department of Aviation.

Johannsen, G., & Rouse, W. B. (1979). Mathematical concepts for modeling human behavior in complex man–machine systems. *Human Factors, 21,* 733–747.

Madhavan, D. (1993). *Cockpit task management errors: An ASRS incident report study.* Unpublished master's thesis, Oregon State University, Corvallis.

Moray, N., Dessouky, M., Kijowski, B., & Adapathya, R. (1991). Strategic behavior, workload, and performance in task scheduling. *Human Factors, 33,* 607–629.

National Transportation Safety Board. (1973). *Eastern Airlines L–1011, Miami, Florida, 29 December 1972* (Report No. NTSB–AAR–73–14). Washington, DC: Author.

National Transportation Safety Board. (1979). *Comair, Inc. Piper PA–31, N6642L, Covington, Kentucky, October 8, 1979* (Report No. NTSB–AAR–80–8). Washington, DC: Author.

National Transportation Safety Board. (1981). *Texasgulf Aviation, Inc., Lockheed JetStar N520S near Westchester County Airport, White Plains, New York, February 11, 1981* (Report No. NTSB–AAR–81–13). Washington, DC: Author.

National Transportation Safety Board. (1982). *Air Florida, Inc., Boeing 737–222, N62F, collision with 14th Street Bridge, near Washington National Airport, Washington, DC, January 13, 1982* (Report No. NTSB/AAR–82–08). Washington, DC: Author.

National Transportation Safety Board. (1986a). *China Airlines B–747–SP, N4522V, 300 nautical miles northwest of San Francisco, California, February 19, 1985* (Report No. NTSB/AAR–86–03). Washington, DC: Author.

National Transportation Safety Board. (1986b). *Piedmont Airlines Flight 467 Boeing 737–222, N752N Charlotte Douglas International Airport, Charlotte, North Carolina, October 25, 1986* (Report No. NTSB/AAR–87–08). Washington, DC: Author.

North, R. A., & Riley, V. A. (1989). W/INDEX: A predictive model of operator workload. In G. R. McMillan, D. Beevis, E. Salas, M. H. Strub, R. Sutton, & L. Van Breda (Eds.), *Applications of human performance models to system design* (pp. 81–89). New York: Plenum.

Raby, M., & Wickens, C. D. (1994). Strategic workload management and decision biases in aviation. *The International Journal of Aviation Psychology, 4,* 211–240.

Steel, R. G. D., & Torrie, J. H. (1980). *Principles and procedures of statistics: A biometrical approach.* New York: McGraw-Hill.

Stewart, S. (1992). *Flying the big jets* (3rd ed.). Stillwater, MN: Specialty Press.

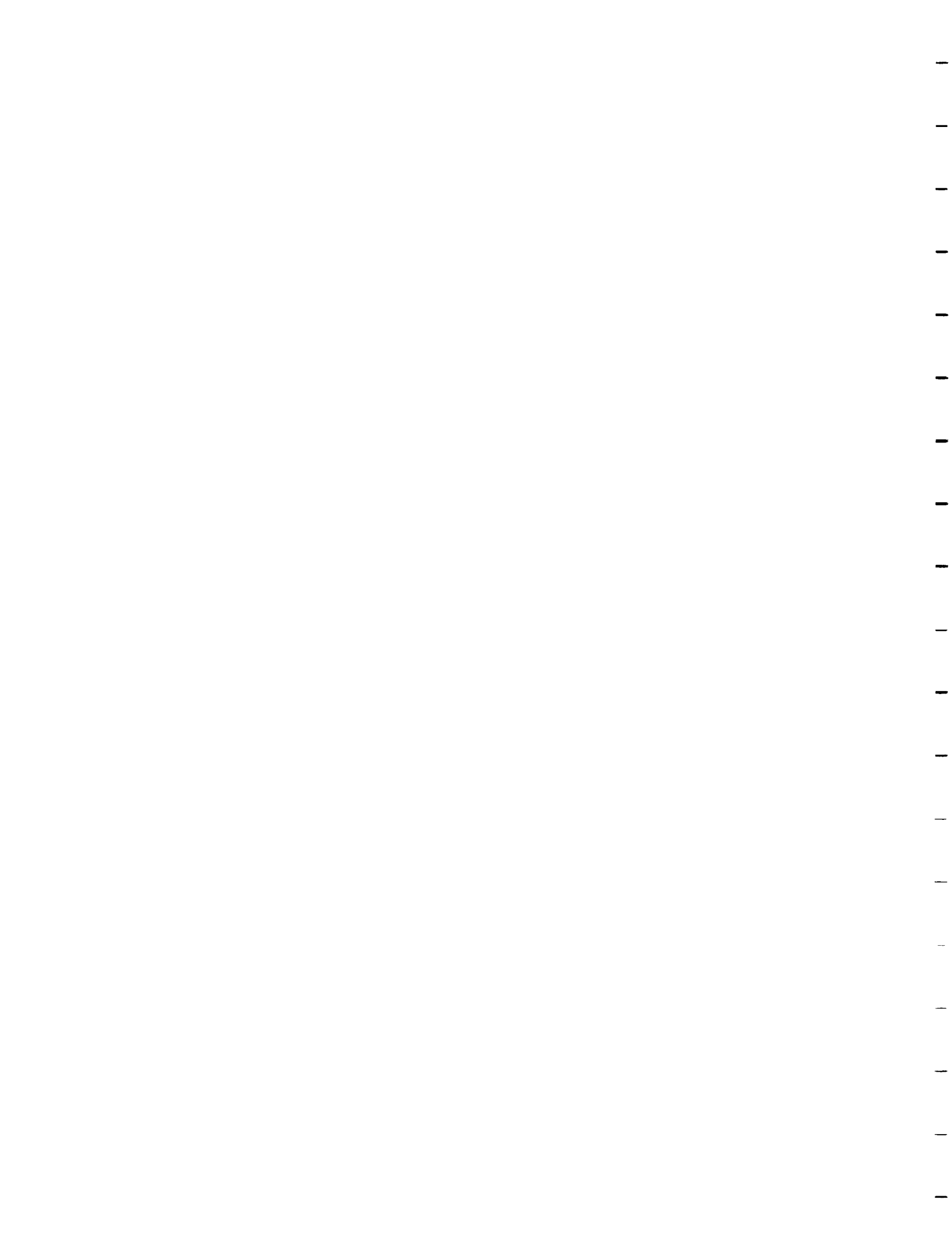Wickens, C. D. (1992). *Engineering psychology and human performance.* New York: HarperCollins.

An Aid to Facilitate Cockpit Task Management Performance

Joong Nam Kim
RothWell International
P.O. Box 270566
Houston, TX 77277-0566
USA
713-660-8080
jkim@rwi.com

Ken Funk
Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon 97331-2407
USA
503-737-2357
funkk@engr.orst.edu

AN AID TO FACILITATE COCKPIT TASK MANAGEMENT PERFORMANCE

Joong Nam Kim
RothWell International
Bellaire, Texas
USA

Ken Funk
Oregon State University
Corvallis, Oregon
USA

INTRODUCTION

In an aircraft cockpit, the pilot performs multiple, concurrent tasks to accomplish the flight mission. For example, the pilot may simultaneously lower the landing gear and communicate with air traffic control (ATC) while maintaining a correct descent rate. The pilot has two principal cockpit roles: controller and manager. Like a driver in an automobile, the pilot as controller performs operational-level tasks such as moment-to-moment manual control and activation/deactivation of automatic devices. As system manager, like a factory manager, the pilot performs such management-level tasks as monitoring system configurations and overseeing activities. In other words, the pilot is in charge of managing the multiple, concurrent flight tasks. Funk (1991) referred to this management-level activity as cockpit task management (CTM).

The basis for the present research follows from prior studies of CTM errors by Funk and his colleagues (Funk, 1991; Chou and Funk, 1993; Madhavan and Funk, 1993; Chou, Madhavan, and Funk, in review). Funk developed a preliminary CTMS theory from the perspective of systems engineering; and Chou (1991) and Madhavan (1993) reviewed aircraft accident and incident reports, verifying the significance of CTM errors in those mishaps. To facilitate CTM and to reduce CTM-related pilot errors, the present study has included the development and evaluation of a prototype aid, the cockpit task management system (CTMS).

BACKGROUND

Cockpit Task Management (CTM)

CTM is "a process by which the flightcrew manages an agenda of cockpit tasks" (Funk, 1991). CTM activities include task

1. initiation,
2. monitoring (i.e, assessing task progress and performance),
3. prioritization,
4. resource allocation , and
5. termination.

CTM Errors

CTM errors occur when a flightcrew fails to perform CTM functions satisfactorily. Chou, Madhavan, and Funk (in review) developed a taxonomy of Cockpit Task Management errors consisting of the error categories

1. task initiation (early, late, or incorrect),
2. task prioritization (incorrect), and
3. task termination (early, late, or incorrect).

They then applied this taxonomy in the analysis of National Transportation Safety Board (NTSB) aircraft accident reports and Aviation Safety Reporting System (ASRS) incident reports. They found CTM errors in 76 (23 per cent) of the 324 accident reports analyzed and in 231 (49 per cent) of the 470 incident reports. They concluded that CTM is a significant factor in flight safety and recommended three general approaches to improve CTM performance: training, procedures, and direct aiding. In regard to the last approach, they recommend that research be conducted to develop and evaluate a computational aid to facilitate CTM performance: a Cockpit Task Management System (CTMS). They recommended that a CTMS

1. maintain a current model of aircraft state and current cockpit tasks,
2. monitor task state and status,
3. compute task priority,
4. remind the pilots of all tasks that should be in progress, and
5. suggest that the pilots attend to tasks that do not show satisfactory progress.

A CTMS can be viewed as an executive associate which would facilitate the pilots' managerial tasks.

Research Objectives

The objectives of the present study were to determine the technical feasibility of a CTMS through the development of a prototype CTMS and to evaluate CTMS effectiveness for the improvement of CTM performance.

METHOD

A CTMS was designed based upon the above requirements. Concepts of object-oriented design (OOD) and distributed artificial intelligence (DAI) were employed in developing the CTMS. The CTMS was then integrated into a PC-based flight simulator for experimental evaluation of system effectiveness. Volunteer subjects flew scenario simulations both with and without the CTMS. Performance data were collected and analyzed to evaluate CTMS effectiveness.

Flight Simulator

The flight simulator used for this research was a small, fixed-based model of an aircraft cockpit for a single pilot, developed by modifying an existing simulator used for a previous CTM study (Chou, 1991). The simulator consisted of three personal computers (each with its own monitor), a computer keyboard, two trackballs, and a sidestick controller. The computers were linked via Ethernet, using the TCP/IP communication protocol.

One of the computers ran a simple aerodynamic model and provided, via its monitor, a primary flight display showing current and command heading, airspeed, and altitude; a pitch ladder indicating pitch and roll angles; aircraft latitude and longitude; and autopilot status (engaged or disengaged).

The second computer and monitor provided a navigation display (ND) consisting of four panels: a horizontal situation indicator (HSI), an automatic flight control (AFC) panel, a source select panel, and an air traffic control (ATC) communication panel. The HSI displayed an aircraft-centered moving map consisting of an aircraft symbol, the current flight path, and waypoint symbols and names. It also displayed the aircraft position, active waypoint data, weather radar data, and an expanded compass rose. The AFC and the source select panels displayed computer-generated button and knob images. These buttons or knobs were used to set the HSI display or a source selector to the desired mode. A trackball was used to both "push" the buttons and "turn" the knobs. The ATC communication panel provided a simplified datalink system for alphanumeric, rather than verbal, communication with a simulated air traffic controller.

The third computer and monitor provided a subsystem display (SD) consisting of six control panels and two display panels. The SD control panels were used to control such aircraft subsystems as the engine, the hydraulic system, and the electrical system, as well as the landing gear and flaps. As for the ND, simulated buttons or knobs in the panels were "pushed" or "turned" using a trackball. The SD display panels provided a simple EICAS (engine indication and crew alerting system) and synoptic displays of aircraft subsystems such as engine, fuel system, hydraulic system, electrical system, and landing gear.

Cockpit Task Management System (CTMS)

Based on the proposed requirements of Chou, Madhavan, and Funk, specific goals for the development of the CTMS were established. These were to help the flightcrew prioritize tasks, initiate tasks, terminate tasks, interrupt tasks, and resume interrupted tasks. To achieve these goals, functional requirements for the CTMS were established. These were to provide the pilot with information about task state, task status, task priority, and task relationships.

The CTMS was implemented on a fourth networked personal computer using Smalltalk, an object-oriented programming language. Concepts of object-oriented design (OOD) and distributed artificial intelligence (DAI) were employed in CTMS implementation.

The CTMS is a knowledge-based system in which problem-solving knowledge is distributed among software units referred to as "agents." Simulated aircraft subsystems and pilot tasks are represented in the CTMS by "system agents" (SAs) and "task agents" (TAs), respectively.

System Agents (SA) An SA is a representative of an aircraft subsystem. A subsystem SA receives state information about its corresponding aircraft subsystem from the flight simulator, releasing this information when requested. In the CTMS, an SA is implemented by an instance of a Smalltalk class, and the specific behaviors or knowledge of the SA are implemented in the methods (i.e., procedures) of the class.

Task Agents (TA) Task agents are responsible for monitoring the performance of corresponding flight tasks. Like SAs, TAs are implemented by an instance of a class, and the specific behavior or knowledge necessary for each TA is implemented in the methods of the class. This knowledge allows each TA to determine when its task should be started, when it should be terminated, and how its status (performance -- satisfactory or unsatisfactory) should be assessed.

CTMS Operations As a subject controls the simulator, CTMS SAs maintain knowledge about the current state of the simulated aircraft, providing that knowledge to TAs on demand. TAs in turn determine when tasks should be started and stopped and continually assess the status of each task. Higher-level TAs prioritize tasks based on a pre-defined priority scheme and identify tasks requiring pilot attention.

From the pilot's perspective, the core unit of the CTMS is its display, which provides information about all tasks with respect to task state, status, and priority. This color alphanumeric display consists of three sections. The upcoming task display (UTD) lists those tasks that should be started soon (e.g., an upcoming descent). The in-progress task display (ITD) lists those tasks that should actually be in progress. The suggested task display (STD) lists any tasks that require immediate attention due either to poor performance or urgency. With this three-segment arrangement, task state (upcoming, in-progress, or suggested) is presented using location coding.

Task status (satisfactory performance or unsatisfactory performance) is indicated by the use of color coding. That is, if the performance on a task is satisfactory, its name is displayed in green; if it is unsatisfactory, a yellow or red color is used, depending upon the importance of the task.

In addition to task state and status, task priority information is presented on the STD, the topmost of the three CTMS display sections. That is, names of the suggested tasks (those needing immediate attention) are listed in order of the priority of the tasks, with higher priority tasks being placed higher in the list.

Experiment

After the CTMS was implemented, an experiment was performed to evaluate its effectiveness in improving CTM performance. Twelve volunteer subjects were used for the experiment. The first four subjects were used for a pilot study to check the readiness of the experiment and facilitate final refinements, and the remaining eight subjects were used for the data collection runs.

A balanced experimental design was developed for the data-collection flights. To compare subject performances between flying with and without the CTMS, each subject flew two data-collection scenarios -- one with the CTMS and the other without it.

Two different flight scenarios were developed to remove the learning effect that would have resulted from the use of the same scenario in the two data-collection flights. Each was designed to present the same complexity to minimize any effect of differences in scenario complexity, which could bias the results of the experiment.

The experimental procedure was administered in two sessions: a training session and a data-collection session. After a four-hour first-day training session followed by a two-hour second-day training session, each subject flew two 50-minute data-collection flight scenarios with a 5 to 10 minute break between flights.

Four measurements were considered for the evaluation of subject performance in the flight simulator:

1. task prioritization (i.e., correct or incorrect),
2. pilot response time (e.g., to equipment faults),
3. task completion (i.e., whether or not a task was completed), and
4. aircraft control (i.e., deviation from planned flight path).

The first three of the four measurements reflected the three elements in the CTM error taxonomy discussed above: task prioritization, task initiation, and task termination, respectively. Subject performances in the use of aircraft controls, including heading, altitude, and airspeed controls, were measured because they were essential to the comprehensive measurement of overall pilot performance. Subject performance from the 16 scenario flights flown by eight subjects was collected for these four performance measures. Simulator log files, which recorded pilot actions and performance measures, as well as videotapes were used to collect performance data.

RESULTS

In association with the four performance measurements described above, data for the following four variables were collected: the ratio of misprioritizations to opportunities for misprioritization, the time required for subjects to first respond to unsatisfactory flight tasks, the proportion of unsatisfactory aircraft control time during a flight, and the total number of flight tasks the subjects failed to complete by the end of the flights.

One of the goals of the research was to determine if the CTMS provided effective flight task assistance during simulated flights. To arrive at this determination, mean subject performances flying with and without the CTMS were compared. As shown in Figure 1, when subjects flew with the assistance of the CTMS, the mean task misprioritization rate was reduced by 41 per cent, the mean subject response time was reduced by 18 per cent, the exercise of mean unsatisfactory aircraft controls was reduced by 24 per cent, and the average number of incomplete tasks during simulator flights was reduced by 82 per cent.

In addition to comparing the subject performance averages, a statistical analysis of the collected data using an analysis of variance (ANOVA) was performed as an additional means of determining whether use of the CTMS resulted in improved subject performance. Since the hypothesis test using the ANOVA was based upon the expectation that performances with the CTMS would be better than performances without the CTMS, a one-tailed test was employed. A type I error, denoted by α, for both 0.1 and 0.05, was used insofar as this form has gained acceptance for use in typical statistical analyses. In such analyses, the results of a hypothesis test are reported as a number called the "p-value" -- a measurement of the credibility of the hypothesis test. A type I error probability, α, and a p-value are used to determine whether the null hypothesis, denoted by H_o, can be rejected. Since the principal concern of this experiment was CTMS effectiveness, as indicated by the p-values for the treatment effect, only these values are presented (Table 1).
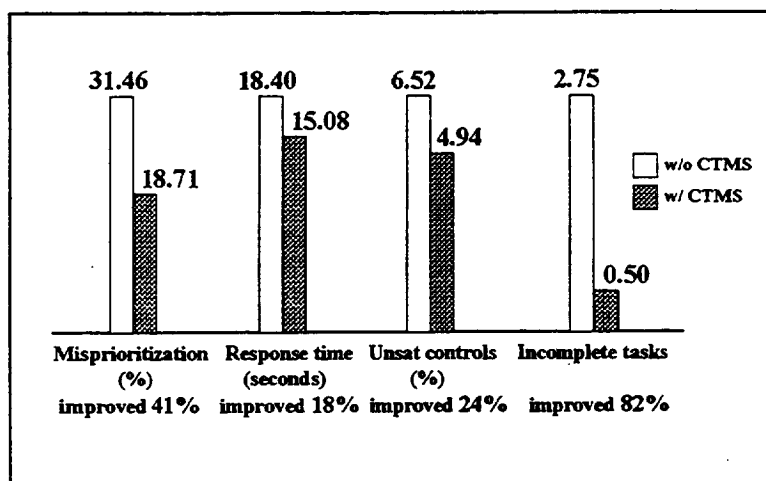
Figure 1. Mean subject performance for flights with and without CTMS assistance.

Performance measure	p-value	Conclusion, $\alpha=0.1$	Conclusion, $\alpha=0.05$
Misprioritization	0.066	reject H_0	do not reject H_0
Response time	0.093	reject H_0	do not reject H_0
Aircraft controls	. 0.052	reject H_0	do not reject H_0
Incomplete tasks	0.009	reject H_0	reject H_0

Table 1. ANOVA p-values for treatment effect and hypothesis test results.

From the results of the hypothesis test, the p-value for incomplete tasks indicated that there was significant improvement for task completion performance when subjects flew with the assistance of the CTMS, whereas the p-values for the remaining three measurements for task prioritization, task initiation, and aircraft controls indicate that there is suggestive evidence of performance improvement.

DISCUSSION

These results indicate that the CTMS was effective in improving CTM performance under the experimental conditions. In other words, they show that if an aid can accurately determine what tasks the pilot is attempting to complete and how well the tasks are being performed, CTM performance can be facilitated by displaying relevant task management information, in particular, calling the pilot's attention to tasks which are not being performed in a satisfactory or timely manner. That the CTMS was successful in this is due in large part to the simplicity of the simulated aircraft, environment, and tasks.

Nevertheless, the findings do point to the potential benefit of such an aid. Past and ongoing research in intent inferencing (Hoshstrasser and Geddes, 1989), activity tracking (Callantine and Mitchell, 1994), and hazard monitoring (Skidmore et al, in press) in higher fidelity environments indicates that at least some of the benefits accruing from the CTMS under laboratory conditions may well be obtainable in more realistic environments.

ACKNOWLEDGEMENT

REFERENCES

Callantine, T.J. and Mitchell, C.M. (1994). A methodology for understanding how operators select and use modes of automation to control complex dynamic systems, in *Proceedings of the 1994 International Conference on Systems, Man, and Cybernetics*. Piscataway, NJ: The Institute of Electrical and Electronics Engineers.

Chou, C.D. (1991). *Cockpit task management errors: a design issue for intelligent pilot-vehicle interfaces.* Unpublished doctoral dissertation, Oregon State University.

Chou, C.D. and Funk, K. (1993). Cockpit task management errors in a simulated flight operation, in *Proceedings of the Seventh International Symposium on Aviation Psychology* (pp. 965-969). Columbus, Ohio: Department of Aviation, The Ohio State University.

Chou, C.D., Madhavan, D. and Funk, K. (in review). Studies of cockpit task management errors. *The International Journal of Aviation Psychology*.

Funk, K. (1991). Cockpit task management: preliminary definitions, normative theory, error taxonomy, and design recommendations. *The International Journal of Aviation Psychology*, 1(4), 271-285.

Hoshstrasser, B.H. and Geddes, N.D. (1989). OPAL: Operator intent inferencing for intelligent operator support systems, in *Proceedings of the IJCAI-89 Workshop on Integrated Human-Machine Intelligence in Aerospace Systems* (pp. 53-70). Minneapolis: Honeywell Systems and Research Center.

Madhavan, D. (1993). *Cockpit Task Management Errors: an ASRS Incident Report Study*. Unpublished master's thesis, Oregon State University.

Madhavan, D. and Funk, K. (1993). Cockpit task management errors in critical in-flight incidents, in *Proceedings of the Seventh International Symposium on Aviation Psychology* (pp. 970-974). Columbus, Ohio: Department of Aviation, The Ohio State University.

Skidmore, M.D., J.P. Zenyuh, R.L. Small, T.J. Quinn, and Moyers, A.D. (in press). Dual use applications of hazard monitoring: commercial and military aviation and beyond , in *Proceedings of the IEEE National Aerospace and Electronics Conference, NAECON 95*. Piscataway, NJ: The Institute of Electrical and Electronics Engineers.

Agent-Based Aids to Facilitate Cockpit Task Management

Ken Funk
Industrial and Manufacturing Engineering
Oregon State University
Corvallis, Oregon, USA 97331-2407
503-737-2357 / funkk@engr.orst.edu

Joong Nam Kim
RothWell International
P.O. Box 270566
Houston, Texas, USA 77277-0566
713-660-8080 / jkim@rwi.com

1. Introduction

In today's highly automated aircraft, the role of the pilot has changed from an airplane controller to a system manager. As a system manager in a cockpit, today's pilot is in charge of a supervisory activity we call cockpit task management (CTM). CTM activities include the initiation, assessment, prioritization, execution, and termination of tasks.

This paper describes our past and ongoing efforts to understand CTM and to facilitate it through the use of agent-based, computational aids.

2. The Task Support System

We became aware of the need for a concept such as CTM as we developed the Task Support System (TSS), part of an experimental avionics package for a military aircraft [1].

Objectives
The purpose of the TSS was to help military pilots execute tasks quickly and correctly. But as we worked on the TSS, we found that it was just as important to help the pilot manage tasks, for even a well-performed task does little towards mission success if it is the wrong task or if higher priority tasks are neglected.

Simulator Environment
We developed the TSS for a simulator representing a single-seat military aircraft and its tactical environment. The simulator was implemented on a Silicon Graphics Iris computer.

Architecture and Implementation
Due to the complexity of the cockpit environment, we used methods of distributed artificial intelligence to implement the TSS. Its major components were intelligent agents: software modules which represented significant elements of the cockpit and its environment, having adequate declarative and procedural knowledge to deal with subsets of the problem domain. System agents represented aircraft subsystems. They monitored subsystem data and maintained declarative knowledge about the aircraft and its environment for other parts of the TSS. Task agents represented cockpit tasks. Each task agent had a knowledge base that helped it determine when a task should be performed and how to work cooperatively with the pilot to complete it successfully. High level task agents used their knowledge bases to help prioritize tasks. We implemented the TSS on an 80386-based personal computer in Objective-C, an object-oriented superset of the C programming language.

Evaluation
We evaluated the TSS in a simulator experiment in which 16 military pilots flew simulated missions in both baseline (no TSS) and enhanced (with TSS) cockpits. With the enhanced cockpit, overall task performance improved 38%, workload (as measured by NASA-TLX) was reduced by 13%, and pilot-perceived effectiveness improved by 83%. 81% of the pilots preferred the enhanced cockpit to the baseline.

Limitations
Although most improvements were statistically significant, conclusive evidence of the success in improving CTM performance, especially in an operational setting, was not demonstrated. First, at that time, we had established no objective measures of CTM performance. Second, as the TSS was not the only element in the enhanced cockpit, it was not possible to separate out its effects. Third, the simulator environment was of low fidelity and the possibility of successful integration of the TSS into an operational aircraft was by no means assured.

3. The Cockpit Task Management System

Cockpit Task Management
Following the successful demonstration of the TSS, we formalized the notion of CTM around the following concepts[2]. A goal is a desired aircraft or system state. A task is a process to achieve a goal. CTM is the process of initiating, monitoring, prioritizing, and terminating tasks. Next we began studies of CTM in the commercial aviation domain to determine its significance to flight safety.

We developed a CTM error taxonomy and, in a study of 324 US National Transportation Safety Board aircraft accident reports, found CTM errors in 29 per cent of the accidents [3, 4]. Using a simplified error taxonomy, in an analysis of 470 Aviation Safety Reporting System incident reports, we found CTM errors in almost 50 percent of the incidents [4, 5]. Concluding that CTM was indeed a significant factor in flight safety, we developed a prototype pilot vehicle aid called the Cockpit Task Management System (CTMS) to facilitate CTM [6].

Objectives

The objectives of this part of our study were to determine the feasibility of CTMS implementation through the development of a prototype CTMS and to evaluate CTMS effectiveness in the improvement of CTM performance.

Simulator Environment

The flight simulator used for this research was a small, fixed-based model of an aircraft cockpit for a single pilot. We developed it by modifying the existing flight simulator used for a previous CTM study [2]. The simulator consisted of three personal computers, each with its own monitor, a computer keyboard, two trackballs, and a sidestick controller. All of the simulator computers were linked via Ethernet, using the TCP/IP communication protocol.

The top monitor was a simulated head up display (HUD) showing aircraft heading, airspeed, and altitude; a pitch ladder; aircraft horizontal location; and autopilot status (i.e. engaged or disengaged). The bottom left monitor, called the navigation display (ND), showed aircraft horizontal position on a moving map display and provided a simple datalink system for simulated air traffic control (ATC) communication. The subsystem display (SD) on the right monitor showed synoptic displays of several simulated aircraft subsystems such as engines, a hydraulic system, and an electrical system. It also provided an interface for a simple flight path management system and warning and alerting displays.

Architecture and Implementation

Our goals for the CTMS were that it should help the pilot initiate, monitor, prioritize, and terminate tasks. To achieve these goals, we determined that the CTMS should provide information about task state (upcoming, active, terminated), status (satisfactory or unsatisfactory performance), and priority.

We implemented the CTMS using Smalltalk, an object-oriented computer programming language. As for TSS development, we used concepts of object-oriented design and distributed artificial intelligence in the CTMS implementation, where aircraft subsystems and flight tasks were represented by conceptual software units referred to as agents. In the CTMS, aircraft subsystems and pilot tasks were represented by system agents (SAs) and task agents (TAs), respectively. The CTMS was an object-oriented, agent-based system in which problem-solving knowledge was distributed among SAs and TAs.

System Agent (SAs): An SA was a representative of an aircraft subsystem. A subsystem SA received state information about its corresponding aircraft subsystem from the flight simulator, releasing this information when requested. For the CTMS, an SA was implemented as an instance of a class, and the specific behaviors or knowledge of the SA were implemented in the methods of the class. Table 1 provides a partial list of simulated aircraft subsystems and the corresponding Smalltalk SA classes.

Aircraft Subsystem	Class
airframe	AirframeAgent
left, right engine	EngineAgent
hydraulic system	HydraulicSystemAgent
autopilot	AutopilotAgent
electric power system	ECSAgent
fuel system	FuelSystemAgent
landing gear	LandingGearAgent
flaps	FlapAgent
electrical input unit	EIUAgent
flight director	FDAgent
inertial reference system	IRSAgent
navigation computer	NavigationAgent

Table 1. Partial list of simulator subsystems and CTMS SA classes.

Task Agents (TAs)

Task agents were responsible for helping the pilot perform corresponding flight tasks. Like SAs, TAs were implemented as instances of Smalltalk classes, and the specific behavior or knowledge necessary for each TA was implemented in the methods of the class. Table 2 provides a partial list of flight tasks and the corresponding Smalltalk TA classes.

CTMS Operation

Each TA used information from SAs and its own procedural knowledge to determine the state of its task: latent (not imminent), upcoming (imminent), in-progress, suggested (requiring immediate attention), or finished. Task status (satisfactory or unsatisfactory) was determined in a similar way.

Flight Task	Class
climb	ClimbAgent
cruise	CruiseAgent
descent	DescentAgent
approach	ApproachAgent
land	LandAgent
fly_to_a_position	FlySegmentAgent
maintain_heading	ManageControlAgent
maintain_altitude	ManageControlAgent
maintain_airspeed	ManageControlAgent
maintain_flap	ManageControlAgent
manage_contingency	ManageContingencyAgent

Table 2. Partial list of flight tasks and CTMS TA classes.

The CTMS display provided information about all tasks with respect to the following four characteristics: (a) state, (b) status, (c) priority, and (d) task-subtask relationship. The display consisted of three sections: (a) UTD (upcoming task display), (b) ITD (in-progress task display), and (c) STD (suggested task display), with task information provided on the corresponding display sections. That is, task state information was presented using location coding.

Task status was either *satisfactory* or *unsatisfactory* and indicated by the use of color coding. That is, if a task was being performed satisfactorily, a green color was used; if its performance was unsatisfactory, a yellow or red color was used, depending upon severity.

In addition to task state and status, task priority information was presented on the STD. That is, names of the *suggested* tasks were displayed according to the priority of the tasks, with higher priority tasks being placed higher in the list. Using the UTD or ITD, the *upcoming* and *in-progress* tasks, respectively, were displayed in hierarchical structure.

Experiment
After the CTMS was implemented and interfaced to the flight simulator, an experiment was performed to evaluate its effectiveness in improving CTM performance. Twelve volunteer subjects were used for the experiment. The first four subjects were used for a pilot study to check the readiness of the experiment, and the remaining eight subjects were used for the experimental data collection test runs.

We developed a balanced experimental design for the data-collection flights. To compare subject performances between flying with and without the CTMS, each subject flew two data-collection scenarios -- one with the CTMS and the other without it. We developed two different flight scenarios, A and B, to avoid the learning effect that would have resulted from the use of an identical scenario in the two data-collection flights. We designed each to present the same complexity to minimize the effect by the differences in scenario complexity, which could have biased the results of the experiment.

We administered the experimental procedure in two lengthy sessions: a training session and a data-collection session. After a four-hour first-day training session followed by a two-hour second-day training session, each subject flew two 50-minute data-collection flight scenarios with a 5 to 10 minute break between flights.

We used four measurements for the evaluation of subject performance in the flight simulator: (a) task prioritization, (b) pilot response time, (c) aircraft controls, and (d) task completion. Three of the four measurements, task prioritization, pilot response time, and task completion, reflected the three elements in our CTM error taxonomy: task prioritization, task initiation, and task termination, respectively. Subject performance in aircraft control (heading, altitude, and airspeed) was assessed as a comprehensive measurement of overall pilot performance. Pilot performance data from the 16 scenario flights flown by eight subjects were collected for these four performance measures. Simulator log files (containing recorded pilot actions and performance) and videotapes were used to collect performance data.

Results
These files and tapes allowed us to compute four performance measures: (a) the ratio of task misprioritizations to opportunities for misprioritization, (b) the time required for subjects to first respond to unsatisfactory tasks, (c) the proportion of unsatisfactory aircraft control time during a flight, and (d) the total number of tasks the subjects failed to complete by the end of the flights.

As shown in Figure 1, when subjects flew with the assistance of the CTMS, the mean task misprioritization rate was reduced by 41 per cent, the mean subject response time was reduced by 18 per cent, the exercise of mean unsatisfactory aircraft controls was reduced by 24 per cent, and the average number of incomplete tasks during simulator flights was reduced by 82 per cent.

In addition to comparing the subject performance averages, we performed a statistical analysis of the collected data using an analysis of variance (ANOVA). Since the hypothesis test using the ANOVA was based upon the expectation that performances with the CTMS

would be better than performances without the CTMS, we employed a one-tailed test. We considered the probability of a type I error, denoted by α, of both 0.1 and 0.05, insofar as this form has gained acceptance for use in typical statistical analyses. In such analyses, the results of a hypothesis test are reported as a number called the *p-value* -- a measurement of the credibility of the hypothesis test. A type I error probability, α, and a p-value are used to determine whether the null hypothesis, denoted by H_o, can be rejected. Since the principal concern of this experiment was CTMS effectiveness, as indicated by the p-values for the treatment effect, we present only these values in Table 3.
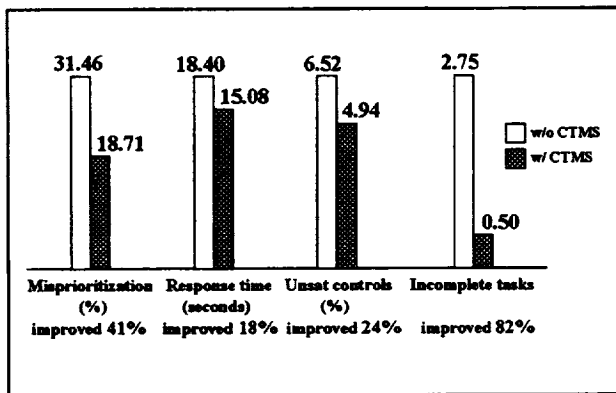

Figure 1. Mean subject performance with and without CTMS assistance (normalized).

Measure	p	For α=0.1 conclude	For α=0.05 conclude
Misprioritization	0.066	reject H_o	do not reject H_o
Response time	0.093	reject H_o	do not reject H_o
Aircraft controls	0.052	reject H_o	do not reject H_o
Incomplete tasks	0.009	reject H_o	reject H_o

Table 3. ANOVA p values for treatment effect and hypothesis test results. H_o is the hypothesis that the CTMS did not improve performance.

From the results of the hypothesis test, the p-value for incomplete tasks indicated that there was significant improvement for task completion performance when subjects flew with the assistance of the CTMS, whereas the p-values for the remaining three measurements for task prioritization, task initiation, and aircraft controls were suggestive with respect to evidence of performance improvement.

Limitations
These results indicate that the CTMS was effective in improving CTM performance under the experimental

conditions. In other words, they show that if an aid can accurately determine what tasks the pilot is attempting to complete and how well the tasks are being performed, CTM performance can be facilitated by displaying relevant task management information, in particular, calling the pilot's attention to tasks which are not being performed in a satisfactory or timely manner. That the CTMS was successful in this is due in large part to the simplicity of the simulated aircraft, environment, and tasks.

4. The AgendaManager

Agenda Management
Recent developments and events require that we broaden the concept of CTM to address issues that are now arising in commercial aviation. First, human pilots are no longer the only actors in the cockpit. Autopilots, thrust management computers, and flight path management systems are playing more and more active roles in the control of advanced technology commercial aircraft. Like human actors, these machine actors are goal-directed systems that use complex data or knowledge bases to determine their behaviors.

As the term *task* is often reserved only for functions performed by humans [7], it now seems to us wiser to call a process performed to achieve a goal a *function* rather than a task. Therefore the management of activities in the modern cockpit must address both human and machine functions.

Second, several recent accidents involving advanced technology aircraft have been due in part to human actors (pilots) working at cross-purpose with machine actors (autopilots). In other words, goal conflicts between actors -- especially when the human actors were not aware of the conflicts -- contributed to event sequences leading up to the accidents.

Based on these two considerations and the understanding that these concepts apply beyond just the cockpit environment, we have chosen to extend our study of Cockpit Task Management to that of *Agenda Management*, the management of goals and functions, the actors who perform those functions, and the resources that they use.

Objectives
The objectives of our current efforts are to develop and evaluate an aid to facilitate Agenda Management in a simulated cockpit environment of higher fidelity than that of either the TSS or CTMS. We refer to the aid, now under development, as the AgendaManager.

Simulator Environment

The environment for the AgendaManager is a part-task simulator based on -- and using software components from -- the Advanced Concepts Flight Simulator (ACFS) of the Man-Vehicle Systems Research Facility at the NASA Ames Research Center. The ACFS is a full-cockpit, motion-base simulator that models a hypothetical two engine turbojet transport with all-electronic displays and autoflight and flight management systems based on those of current Boeing aircraft.

The part-task version of the ACFS we are developing runs on one or two Silicon Graphics Indigo 2 computers and provides a simplified aerodynamic model, autoflight system, primary flight displays, and system synoptic displays. The software is being written in C and Smalltalk. Use of selected, non-proprietary ACFS components (provided by NASA Ames) and conformance to ACFS functionality and communication protocols assures an environment migration path to the full ACFS for the AgendaManager.

Functional Requirements

As a first step in designing the AgendaManager, we developed a formal, functional model of Agenda Management using IDEF$_0$, a graphical modelling methodology useful for representing and decomposing complex activities.

As the IDEF$_0$ model itself is too large to include in this paper, Table 4 presents just the top-level activities -- functions themselves -- of Agenda Management. Each activity is represented by an IDEF$_0$ node identifier, consisting of the letter 'A' (for Activity) and a sequence of digits coding subordination relationships (e.g., A12 is the second sub-activity of activity A1). The identifier is followed by the name of the activity (a verb phrase) and a short definition. The activities marked with an asterisk (*) define the functional requirements of the AgendaManager: these are activities that the AgendaManager must perform or assist the flightcrew in performing.

Architectural Elements

Next, from the IDEF$_0$ model we generated a data dictionary consisting of the entities that are the inputs and outputs (products) of the activities in the model. These helped us identify necessary components of the AgendaManager's architecture. Major elements include System Agents, Actor Agents, Goal Agents, Function Agents, and Agenda Agents. Each agent will represent the corresponding entity in the cockpit environment and be implemented as a software object.

A0* perform flightdeck activities: Perform the activities of operating a commercial transport aircraft from its flightdeck. These activities are performed by human actors (flightcrew) and machine actors (flightdeck automation) using flightdeck resources (displays, sensors, controls, actuators, radios, and other non-'intelligent' devices).

A1* manage agendas: Manage the actors' agendas. An agenda consists of a set of goals, a set of functions to achieve these goals, a set of actor assignments, and a set of resource allocations.

A11* manage individual agendas: Manage the agenda of each individual actor.

A111* manage goals: Recognize, infer, activate, and terminate goals. Recognize and resolve goal conflicts. Prioritize active goals.

A112* manage functions: Initiate, assess, prioritize, and terminate functions to achieve goals.

A113 assign actors to functions: Decide which actors perform which functions.

A114 allocate resources to functions: Decide what resources to use for each function.

A12* share agenda information: Share information about agendas among actors.

A2* perform other functions: Perform specific functions to achieve mission goal and subgoals.

A21 coordinate actors: Coordinate the activities of the actors assigned to each function.

A22* assess function: Assess the status of each function: how well it is being performed and the likelihood that the goal will be achieved.

A23 maintain situation models: Integrate new situation information to update the situation models.

A24 decide/plan: Decide on what actions to perform immediately or in the future.

A25 act: Transform the decisions into actions.

Table 4. Top-level activities from a functional model of Agenda Management.

System Agents Each system agent's declarative knowledge will include the past, current, and projected future state of the corresponding system as well as that system's status (normal or abnormal). Its procedural knowledge will include how to obtain and project state and status information.

Actor Agents As declarative knowledge, each actor agent will maintain information about the current state of the corresponding actor, including his/her/its agenda. Actor agent procedural knowledge will cover how to obtain state information. Actor agents for human actors will incorporate intent inferencing as well as explicit goal communication capabilities.

Goal Agents Declarative knowledge for each goal agent will include the state of the goal (pending, active, terminated) and its priority. Its procedural knowledge will include how to assess goal state and compute priority.

Function Agents Each function agent will have declarative knowledge about the state and status of its function and what system agents to monitor to assess its function. Function agent procedural knowledge will include how to assess function state and status.

Agenda Agents An agenda agent will maintain declarative knowledge about its actor's goals and functions. It will have procedural knowledge about how to prioritize goals and functions and how to detect and resolve goal and function conflicts.

We will implement this architecture (again in Smalltalk, but this time on a Silicon Graphics Indigo 2 computer), interface it to the part-task simulator, and evaluate it using flight scenarios under development.

5. Potential Pitfalls

In a separate but related study [8], we and our colleagues at America West Airlines and Honeywell have identified over 100 perceived problems with current cockpit automation. Therefore we are aware that there are dangers of introducing new technology into the cockpit. In particular, the AgendaManager has the potential to increase pilot workload, to induce the erosion of pilot cognitive skills, and (if often but not always effective) invite overconfidence. We therefore have the responsibility to see that these and other issues are addressed in its development.

6. Acknowledgements

7. References

[1] K.H. Funk and J.H. Lind, "Agent-Based Pilot-Vehicle Interfaces," *IEEE Transactions. on Systems, Man, and Cybernetics*, Vol. 22, No. 6, Nov/Dec 1992, pp. 1309-1322.

[2] K. Funk, "Cockpit Task Management: Preliminary Definitions, Normative Theory, Error Taxonomy, and Design Recommendations, *The International Journal of Aviation Psychology*, Vol. 1, No. 4, 1991, pp. 271-285.

[3] C.D. Chou, *Cockpit Task Management Errors: A Design Issue for Intelligent Pilot-Vehicle Interfaces.* Unpublished doctoral dissertation, Oregon State University, 1991.

[4] C.D. Chou, D. Madhavan, and K. Funk, "Studies of Cockpit Task Management Errors," *International Journal of Aviation Psychology*, in review.

[5] D. Madhavan, *Cockpit Task Management Errors: An ASRS Incident Report Study.* Unpublished master's thesis, Oregon State University, 1993.

[6] J.N. Kim, *An Agent-Based Cockpit Task Management System: a Task-Oriented Pilot-Vehicle Interface.* Unpublished doctoral dissertation, Oregon State University, 1994.

[7] R.W. Bailey, *Human Performance Engineering*, second edition. Englewood Cliffs, NJ: Prentice Hall, 1989, p. 181.

[8] K. Funk, B. Lyall, and V. Riley, "Flightdeck Automation Problems," in R.S. Jensen (editor), *Proceedings of the Eighth International Symposium on Aviation Psychology*. Columbus, OH: The Ohio State University Department of Aviation, in press.

A FUNCTIONAL MODEL OF FLIGHTDECK AGENDA MANAGEMENT

Ken Funk and Bill McCoy
Oregon State University
Corvallis, Oregon USA

Our research represents an effort to understand and facilitate the management of flightdeck activities by pilots. We developed a preliminary, normative theory of Cockpit Task Management (CTM) and from it defined an error taxonomy. Based on analyses using this error taxonomy we found CTM errors in 76 (23 per cent) of 324 aircraft accident reports and 231 (49 per cent) of 470 aircraft incident reports. Concluding that CTM is a significant factor in flight safety and recognizing the need to broaden as well as refine the concept, we developed a model of Agenda Management, which includes management not only of tasks, but goals, functions, actor assignments, and resource allocations as well. Major components of the functional model include maintaining situation awareness, managing goals (recognizing, inferring, and prioritizing), managing functions (activating, assessing status, and prioritizing), assigning actors (pilots and flightdeck automation) to functions, and allocating resources (such as displays and controls) to functions.

INTRODUCTION

Pilots of modern aircraft must not only perform multiple, concurrent tasks, they must also manage those tasks as well as other functions being performed by non-human actors on the flightdeck. This paper describes our efforts to understand and facilitate the management of flightdeck activities, a process we call Agenda Management.

Our studies parallel and to a certain extent follow a line of research established by Johannsen and Rouse (1979), Hart (1989), Moray and his colleagues (Moray, Dessouky, Kijowski, & Adapathya, 1991), and Wickens and his colleagues (Raby and Wickens, 1994). The concept common to all of these is that the human operator of a complex system must perform multiple, concurrent tasks to control the system and that, as human perceptual and cognitive resources are limited, must therefore manage those tasks.

Our own efforts to formalize this notion in the context of aviation resulted in a preliminary, normative theory of Cockpit Task Management (Funk, 1991). We defined Cockpit Task Management (CTM) in terms of the following activities:

- task initiation: recognizing that a particular goal must be accomplished and therefore that a task must be performed to achieve it.
- task monitoring: assessing progress towards achieving each goal and the level of performance in executing the task.
- task prioritization: assessing relative task priority in terms of overall mission and safety importance, urgency, and momentum or continuity.

- resource allocation: allocating human and machine resources to the completion of tasks based on task priority.
- task termination: recognizing that a goal is achieved, unachievable, or no longer relevant, and ceasing action on the task.

According to our preliminary theory, these activities comprise a high-level cognitive process which serves to determine which low-level activities (i.e., tasks) are being done at any given time.

To validate the theory, we analyzed aircraft accidents and incidents (Chou, Madhavan, & Funk, in press). First, we developed a CTM error taxonomy consisting of the following error categories:

- task initiation errors: early, late, lacking
- task prioritization errors: incorrect
- task termination errors: early, late, incorrect

We applied this taxonomy to 324 US National Transportation Safety Board (NTSB) aircraft accident reports. These were, to our knowledge, all NTSB reports on aircraft accidents occurring between 1960 and 1989. First we reviewed abstracts of the reports and eliminated those that did not have some clear indication of task management errors. Of the remainder we examined either the abstracts or the complete reports themselves in detail, reinterpreting the NTSB's conclusions in light of CTM theory and the error taxonomy. We found CTM errors in 76 (23 per cent) of the reports.

Next we applied the taxonomy to 470 Aviation Safety Reporting System (ASRS) aircraft incident reports. These

reports were obtained from ASRS in three separate search requests: in-flight engine emergencies, controlled flight toward terrain, and incidents in the terminal phases of flight (descent, approach, and landing). We reviewed the narrative sections of these reports, where the reporter describes the incident in his/her own words. We looked for explicit references to neglected tasks, misprioritizations, and delays, again interpreting the conclusions of the reporter in terms of CTM theory. We found CTM errors in 231 (49 per cent) of the reports.

From these studies we concluded that CTM is a significant factor in flight safety and thereby warrants both further study and efforts to facilitate it to reduce the likelihood of error.

Events subsequent to these studies, in particular, several accidents involving highly automated aircraft, have led us to change our perspective, definitions, and terminology somewhat.

In particular, if we define an actor as an entity capable of goal-directed activity, it is very clear that human pilots are not the only actors in the cockpit or on the flightdeck. Monitoring and control of the aircraft and its subsystems are performed by machine actors as well, such as autopilots, flight management systems, and automated warning and alerting systems. A common definition of task is a function performed by a human, where a function is a process performed to achieve a goal. Therefore, we must acknowledge that flightcrews in automated aircraft manage functions, not just tasks.

Furthermore, it is also clear, especially from some recent accidents, that actors frequently have conflicting goals, and that these conflicts may lead to conflicting actions, resulting in unsafe conditions. Goals must be managed too.

From these insights, we have changed our terminology and now refer to Agenda Management. An agenda is a set of goals, functions, actor assignments, and resource allocations. Managing this agenda is an important process performed by the flightcrew.

OBJECTIVES

The objectives of our research are to

1. develop and validate a formal model of Agenda Management.
2. investigate means of facilitating Agenda Management.

METHOD

Since Agenda Management is an activity or a function itself, we decided that a functional modeling approach would be appropriate. We performed a functional decomposition of the process using IDEF0, a graphical modeling tool. An IDEF0 model consists of block diagrams representing activities or functions that transform entities,

and the entities those functions act on or are constrained by. The functions are denoted by verb phrases, the entities are denoted by noun phrases. IDEF0 provides a framework that helps the modeller identify key transformations that take place, the objects of the transformations, factors which limit or guide the transformations, and the mechanisms that perform the transformations.

Starting at the most general level of flightdeck activities, we used knowledge derived from the studies described above to decompose higher level activities to lower level activities, continuing to a level we felt was necessary for validation and adequate to help guide the development of Agenda Management aids.

RESULTS

A portion of our functional model of Agenda Management is presented below. In particular, major functions in the process are identified and defined. Each function is denoted by its IDEF0 identifier, which consists of the letter 'A' (for Activity) and a sequence of digits showing hierarchical relationships between functions (A11 is the first subfunction of A1, A112 is the second subfunction of A11, etc.).

A0 perform flightdeck activities -- Perform the activities of operating a commercial transport aircraft from its flightdeck. These activities are performed by human actors (flightcrew) and machine actors (flightdeck automation) using flightdeck resources (displays, sensors, controls, actuators, radios, and other non-'intelligent' devices). The actors may be viewed as a single, integrated cognitive system.

- **A1 manage agendas** -- Manage the agendas of all actors.

  - **A11 manage individual agendas** -- Manage the agenda of each individual actor. Each actor manages his/her/its own agenda and these agendas may or may not be consistent.

    The following subfunction descriptions (A111 through A1144) reflect the activities performed by a single actor in the management of his/her/its own agenda.

    - **A111 manage goals** -- Recognize, infer, activate, and terminate goals. Prioritize active goals. This must be coordinated with the goal management of other actors through shared agenda information.

- **A1111 infer goals** -- Infer the other actors' goals from actor and other system state information in the situation models: "What are the other actors' goals that they have not explicitly declared?"

- **A1112 assess goals** -- Determine what goals should be pursued. Initially, this is just the mission goal, which is decomposed into subgoals. But at any given time, this activity involves adding goals inferred from other actors and this actor's newly derived goals to the set of current (pre-existing) goals, then assessing each to determine if it is pending, active, or terminated: "What should we be getting ready to do (pending goals)? What should we be doing now (active goals)? What can we forget about (terminated goals)?"

- **A1113 prioritize goals** -- Rank the goals based on the importance and urgency of each goal. A goal has high importance if its achievement is a necessary condition for achieving the mission goal. It has high urgency if it must be achieved soon. "What is most important? What is most urgent? What is most worthy of our attention right now?"

- **A1114 identify goal faults** -- Identify any goal problems, such as erroneous or conflicting goals: "Are our goals appropriate and are we in agreement about them?"

- **A112 manage functions** -- Initiate, assess, prioritize, and terminate functions to achieve goals. This must be coordinated with the function management of other actors through shared agenda information.

  - **A1121 activate/deactivate functions** -- Based on the active goals, determine what functions should be performed now: "Are we actually doing what we should be doing?"

  - **A1122 assess function status** -- Determine how well each function is being performed, with respect to achieving the goal, based on accuracy, speed, and other factors. As well as considering the current state of affairs, look ahead. In addition to using global information, use specific status information derived in the process of performing each function. "How well are we doing now? Are things likely to get better, worse, or stay the same? Is it likely that we will achieve the goals?"

  - **A1123 prioritize functions** -- For each function, determine its priority, based on its goal's priority, its status, and its momentum (i.e., functions nearly completed have a greater momentum than do functions just begun). "What should we be doing right now?"

  - **A1124 identify function faults** -- Identify any problems with the current functions, such as inappropriate functions, misprioritized functions, or discrepancies about functions: "Are we in agreement about what we should be doing right now and how well we're doing?"

- **A113 assign actors to functions** -- Decide which actors are to perform each function. This must be coordinated with the actor assignments of other actors through shared agenda information.

  - **A1131 identify feasible assignments** -- Identify different ways that actors could be feasibly assigned to perform functions: "How could we assign actors to functions?"

  - **A1132 evaluate feasible assignments** -- Evaluate the different ways actors could be assigned to functions: "What are the advantages and disadvantages of particular actor assignments?

  - **A1133 select assignments** -- Select the best actor assignments: "What are the best assignments?"

  - **A1134 identify assignment faults** -- Identify problems with the assignments, such as inappropriate assignments and inconsistencies between actors: "Do we agree on the correct actor assignments?"

- **A114 allocate resources to functions** -- Decide what resources are to be used to perform each function. This must be coordinated with the resource allocations of other actors through shared agenda information.

  - **A1141 identify feasible allocations** -- Identify the feasible ways in which resources could be assigned to functions: "How could we allocate resources to functions?"

- **A1142 evaluate feasible allocations** -- Rate the different feasible allocations: "What are the advantages and disadvantages of different resource allocations?"

- **A1143 select allocations** -- Select the best resource allocation: "What are the best resource allocations?"

- **A1144 identify allocation faults** -- Identify any problems with the resource allocations, such as inappropriate allocations or inconsistencies between actors: "Do we agree on the best resource allocations?"

- **A12 share agenda information** -- Communicate information (overtly and covertly) about agendas among the actors. It is only through sharing agenda information that the individual agendas can approach consistency.

- **A2 perform other functions** -- Perform specific functions (other than managing agendas) to achieve the mission goal and its subgoals. These can include monitoring the state of aircraft subsystems, changing the state of the aircraft and its subsystems by manipulating controls, making decisions, solving problems, and planning. The last function yields additional (derived) goals to accomplish. Performing such functions involves maintaining situation models.

  The following descriptions (A21 through A25) pertain to the performance of a single function to achieve a single goal, possibly by multiple actors.

- **A21 coordinate actors** -- Coordinate the activities of the actors assigned to perform the function. Decide what roles and responsibilities each actor will have in performing the function.

- **A22 assess function** -- Assess the status of this function: how well it is being performed, what the future prospects look like, and the likelihood that the goal will be achieved.

- **A23 maintain situation models** -- Update and exercise each actor's situation model. Each actor has an internal representation of the current state of the world and at least human actors can project their models into the future. Maintenance of these models is driven by the need for performing this function. By extension, similar situation model maintenance activities are conducted in parallel for other functions.

- **A231 determine information requirements** -- Determine what information is needed to perform this function.

- **A232 acquire situation information** -- Obtain information from the environment, the aircraft, and other actors.

- **A233 integrate situation information** -- Integrate new situation information and shared information from other actors' situation models into the current situation models.

  - **A2331 update existing situation information** -- Use new information about the various systems to update their state representations in the situation models.

  - **A2332 add new situation information** -- Add other, new information (not just updates of old information) to the situation models.

  - **A2333 project situation models** -- Use possible courses of action to project the current situation into the future, yielding one or more possible scenarios.

  - **A2334 identify situation model faults** -- Identify problems with the situation models, such as inaccuracies, omissions, internal inconsistencies, and lack of agreement between the models of different actors.

  - **A234 share situation information** -- Communicate about the actors' situation models.

- **A24 decide/plan** -- Decide on what actions to perform immediately to achieve the goal, or plan what to do in the future. Planning may yield subgoals derived from this function's goal. These will be added to the actors' agendas.

- **A25 act** -- Perform the actions necessary to achieve this function's goal. These may include control manipulations, utterances, etc.

DISCUSSION

The model emphasizes that the flightcrew must manage goals and functions, assign actors to functions, and allocate resources to functions. It also underlines the importance of maintaining situational awareness and communicating information about individual agendas to identify and resolve conflicts.

The elements of the full IDEF0 model provide further details to be used in analyzing accident and incident reports to identify where Agenda Management may have broken down. Therefore, it is a potentially useful tool in developing means of facilitating the process of Agenda Management through procedures, training, and computational aids.

MODEL VALIDATION

The full IDEF0 model reflects our understanding of Agenda Management, an understanding built largely from analyzing accident and incident reports and observing subject behavior in our laboratory. It seems to comport well with normal flightdeck operations. However, it must be viewed as a hypothesis, subject to validation.

Our initial approach to validation is a continuation of our incident report studies. We have prepared a list of keywords designed to elicit incident reports in which the reporters describe goals that were not met because a function was not completed or was interfered with due to misprioritizations or other Agenda Management errors.

For each such report we find, we are attempting to determine what goals the flightcrew was pursuing and what functions were not performed satisfactorily as a result of failures in Agenda Management. We will attempt to do a rough quantification of goals and functions in order to ascertain the limits to human Agenda Management performance. We also hope to use the reporters' descriptions to determine if the structure of our model is consistent with flightdeck practice. From this analysis we hope to refine the current model and move towards a model that may be ultimately validated.

We recognize the limitations inherent in incident report studies. Therefore, we anticipate that further validation efforts will involve pilot surveys and simulator experiments.

AIDS TO FACILITATE AGENDA MANAGEMENT

In parallel with our recent modeling efforts, we have been using knowledge gained in our accident and incident studies to develop experimental, computational aids to facilitate Agenda Management (Funk & Kim, 1995). We have learned that if an aid can accurately ascertain flightcrew goals and monitor functions being performed to achieve those goals, it can help improve Agenda Management performance by bringing to the flightcrew's attention goal conflicts, unsatisfactory function performance, and other Agenda Management problems. Our current

efforts center on developing methods for overt and covert goal communication and mechanisms for assessing function performance.

REFERENCES

Chou, C.D., Madhavan, D., & Funk, K. (in press). Studies of Cockpit Task Management Errors. *The International Journal of Aviation Psychology.*

Funk, K. (1991). Cockpit task management: preliminary definitions, normative theory, error taxonomy, and design recommendations. *The International Journal of Aviation Psychology, 1(4),* 271-285.

Funk, K. & Kim, J.N. (1995). Agent-based aids to facilitate cockpit task management. *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics,* Piscataway, NJ: The Institute of Electrical and Electronics Engineers, pp. 1521-1526.

Hart, S.G. (1989). Crew workload management strategies: a critical factor in system performance. In R.S. Jensen (Ed.), *Proceedings of the Fifth International Symposium on Aviation Psychology* (pp. 22-27). Columbus, OH: The Ohio State University, Department of Aviation.

Johannsen, G. & Rouse, W.B. (1979). Mathematical concepts for modeling human behavior in complex man-machine systems. *Human Factors,* 21(6), 733-747.

Moray, N., Dessouky, M., Kijowski, B., & Adapathya, R. (1991). Strategic behavior, workload, and performance in task scheduling. *Human Factors.* 33(6), 607-629.

Raby, M., & Wickens, C.D. (1994). Strategic workload management and decision biases in aviation. *The International Journal of Aviation Psychology, 4(3),* 211-240.
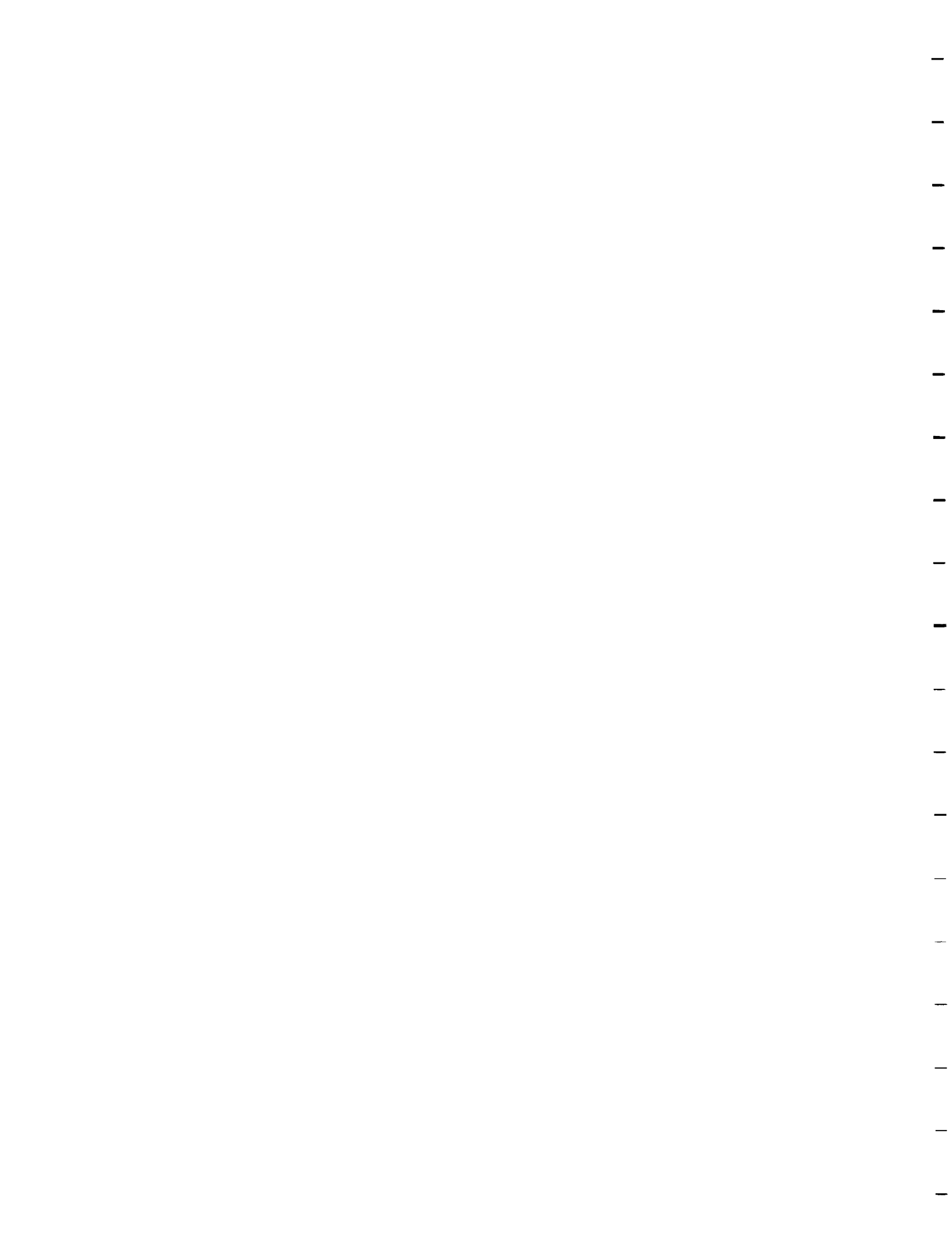
Expanding the Functionality of Existing Airframe Systems Monitors: The AgendaManager
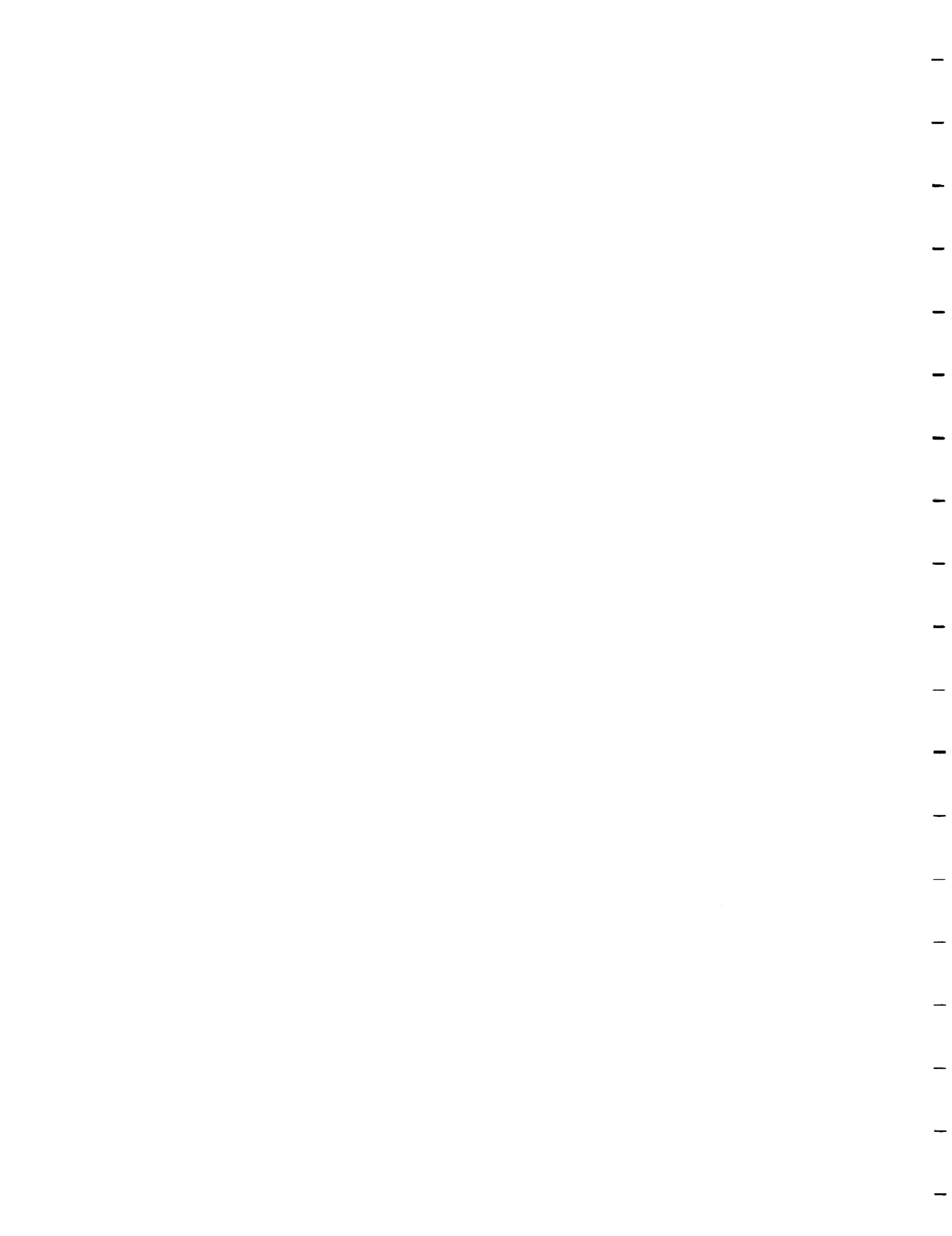
Ken Funk
Dept. of Industrial and Manufacturing Engineering
118 Covell Hall
Oregon State University
Corvallis, OR 97331-2407

541-737-2357
funkk@engr.orst.edu

Rolf Braune
7250 Old Redmond Road, #140
Redmond, WA 98052

206-885-1943
71611.1126@compuserve.com

Expanding the Functionality of Existing Airframe Systems Monitors: The AgendaManager

Ken Funk
Oregon State University
Corvallis, Oregon

Rolf Braune
Redmond, Washington

Background: Existing Systems Monitors

The study of airframe systems monitor concepts and aircraft alerting systems was initiated in 1973 when the Federal Aviation Administration (FAA) contracted with Boeing to study independent altitude monitors (Parks, Hayashi, and Fries, 1973). Follow-on studies conducted during 1974 through 1977 investigated operational philosophies for implementing effective and reliable alerting systems. Study results indicated that there existed a growing proliferation of alerts on the flightdeck and very little standardization had been used by the airframe manufacturers in implementing the alerting system elements. Airline pilots began to view alerting systems as a nuisance rather than a help (Cooper, 1977).

These FAA-funded contracts were performed as joint efforts by the Boeing, Lockheed, and McDonnell Douglas aircraft companies. Data from the first three studies were combined to develop a human factors guidelines document (Boucek, Veitengruber, and Smith, 1977). A second series of studies was conducted and the results combined with the data obtained from the previous investigations to develop the design guidelines contained in an alerting systems guidelines document (Berson et al., 1981). During the course of the contract, interest developed within the FAA in expanding the requirements of the alerting system to monitor overall flight status and facilitate crew responses to non-normal and emergency situations. The results obtained supported the feasibility of expanding the functions of the alerting system to perform as a flight status monitor (FSM). The alerting function of the FSM would serve to alert the flight crew to all non-normal situations for both flight operations as well as aircraft system operations. However, the functional requirements for the FSM were developed on the assumption that, by providing guidance and feedback information, crew performance could be improved.

One of the technologies that benefited directly from the guidelines on crew alerting systems was Boeing's Engine Indicating and Crew Alerting System (EICAS) which was first installed on the Boeing 757 and 767 airplanes (Morton, 1983) and has since been recognized as one of the success stories of flight deck automation (Wiener, 1989). Since then Boeing implemented EICAS on the B747-400 (Boeing, 1989) and the B777-200 (Boeing, 1994). Airbus implemented ECAM on the A320 (Airbus Industrie, 1989) and its family members A319-A340. McDonnell Douglas' Engine Monitor and Display System (EMADS) can be found on the MD-11 as part of the Electronic Instrument System (EIS) alerting system (McDonnell Douglas, 1991)

Independent of its specific implementation as either EICAS, ECAM, or EMADS, the centralized alerting and airplane systems monitoring functions consist of a central crew alerting system and at least one display unit. The following brief description uses Boeing's original 757/767 implementation as example (Boeing, 1988) (Figure 1).

The system consists of a warning electronics unit; two master warning/caution switch lights; discrete alert annunciator lights; and caution and advisory message cancel/recall switches. Three levels of alert messages are presented on the upper display: warnings, cautions, and advisories. The warnings, which are shown in red, are defined as "an operational or aircraft system condition that requires immediate corrective or compensatory action by the crew". Examples of warning level alerts are Fire and Takeoff and Landing Configuration.

The cautions, which are shown in amber, were defined as "an operational or aircraft system condition that requires immediate crew awareness and future compensatory action". The more serious airframe systems malfunctions fall into this category, for example, loss of hydraulic system pressure or autothrottle disconnect. Also shown in amber, but slightly indented, are advisory messages. They are defined as "an operational or aircraft system condition that requires crew awareness and may require corrective action on a time-available basis. For example, autobrakes or doors. Figure 1 shows examples from the B757 representing the three levels of alerts.

Since its introduction on airplanes like the B757/767 the centralized alerting systems have evolved into comprehensive systems which include (1) the airplane's warning system, (2) the engine indication and crew alerting system, (3) the ground proximity system, and (4) the traffic collision and avoidance system. The warning system consists of aural speakers, master warning lights and tactile control column feedback. The aircraft's warning system usually controls and activates visual/aural and/or tactile alerts for warnings like: (1) fire, (2) engine failure, (3) cabin altitude, (4) overspeed, (5) stall warning, (6) takeoff and landing configuration, (7) autopilot disconnect, (8) unscheduled stabilizer movement, (9) ground proximity, (10) windshear, (11) traffic alert and collision avoidance, and (12) crew alertness.

The engine indication and crew alerting system, or electronic centralized aircraft monitor, usually provides (1) system alerts, (2) communication alerts, (3) memo messages, (4) status messages, and (5) maintenance information. The system alert message categories have been expanded from the original three categories (warning, caution, advisory) to include a fourth category, time critical warnings. Time critical warnings are usually associated with primary flight path control. For example, wind shear, terrain/obstacle avoidance, traffic/collision avoidance.

The ground proximity warning system (GPWS) provides alerts for potentially hazardous flight conditions involving imminent impact with the ground. The GPWS also provides an alert for windshear conditions, excessive angle of bank, and glide slope deviations. However, the GPWS does not provide any warning for flight towards vertically sheer terrain or a slow descent into terrain while in landing configuration.

The traffic alert and collision avoidance system (TCAS) alerts the crew to possible conflicting traffic. The system interrogates operating transponders in other airplanes, tracks the other airplanes by analyzing the transponder replies, and predicts the flight paths and positions. Neither advisory, flight path guidance, nor traffic display is provided for other aircraft that do not have operating transponders. Its operation is independent of ground-based air traffic control (ATC).

In order to improve the operational use of the alerting systems the individual alerts are tied to a flight phase logic. This is to avoid distracting alerts especially during high workload phases like takeoff or landing, and also to inhibit them when they are operationally not necessary or inappropriate. For example, Airbus uses ten distinct flight phases for its warning/caution inhibit logic (Airbus, 1989). It uses the same logic for the automatic display of one of the twelve ECAM systems pages on the lower display.

Although the existing centralized indication and alerting systems have generally been very well received by the operational community, they are limited in at least two important areas: (1) ordering and prioritization of information within an alert category, and (2) anticipation of flight crew intent on a moment-by-moment logic. In reference to (1), the key problem is that today's systems during non-normal events display the information in chronological order. If more than one system is affected the resulting possibly long string of messages has to be read and interpreted by the flight crew requiring often extensive systems knowledge in order to ensure a correct diagnosis.

In reference to (2) it could be argued that the Airbus implementation attempts to anticipate the flight crew's intents by tying the display of specific system pages to a flight phase logic. The problem is that this logic is based on a limited set of pre-engineered criteria that do not allow for a moment-to-moment assessment of the actual situation. It is to Airbus' credit that they do allow the pilots to override the automation and select those system pages manually which may be more suitable for the actual situation.

Limitations of Existing Systems Monitors

Although the existing centralized alerting systems have been very well received by the operational community they are limited in the extent to which they can tailor the information to the phase of flight and they are not capable of merging the information in case of multiple failures. Of much greater significance is that little or no effort is made to consider the flightcrew's intent at any given moment.

Existing centralized alerting systems are *system-centered* rather than *function-centered*. That is, they monitor aircraft systems and alert or warn the flightcrew *if and only if* nominal system operating limits are violated, regardless of what functions the pilots are trying to perform to achieve their immediate goals. They do not alert or warn if system parameters are not consistent with pilot intent. For example, existing crew alerting systems will warn the flightcrew of an overspeed condition when the aircraft's maximum operating speed (V_{MO}) is exceeded. But if ATC directs the flightcrew to maintain 240 knots to maintain spacing with other aircraft, existing systems will not advise the pilots if that speed is exceeded because it is not sensitive to their immediate goal.

Furthermore, current alerting systems cannot detect when flightdeck automation (e.g., autoflight or the flight management system) is not configured to operate consistently with flightcrew goals. For example, if the flightcrew has just received an ATC clearance to descend from 12,000 ft to 9,000 ft, and the flight crew intends to comply with the clearance by setting the autoflight system mode control panel (MCP) altitude target, no current alerting system could detect an error and notify the flightcrew if the target altitude value is inadvertently set to 8,000 ft.

The AgendaManager

Building on the successes of existing alerting systems, we are developing and evaluating an experimental, function-oriented monitoring, alerting, and warning system called the AgendaManger (AMgr), which operates in a part-task simulator environment. Consistent with existing crew alerting philosophies, the AMgr monitors system status and alerts and warns the pilot to nominal abnormalities, but the AMgr also monitors systems with respect to current pilot goals. In our part-task simulator, the pilot declares his/her current goals by verbal utterances, drawn mostly from acknowledgments to ATC clearances. It also infers the 'goals' of flightdeck automation (e.g., the target altitude of the autopilot or the active waypoint of the Flight Management System) based on the modes set or parameters programmed by the pilot. The AMgr then monitors flightcrew and system behavior, assessing whether or not those goals are being accomplished satisfactorily. When this is not the case, the AMgr informs the pilot.

This functionality serves several purposes. First, it continually monitors activities to determine if performance is consistent with declared goals. Second, it helps remind the pilot of important tasks that may have been interrupted and not resumed. Third, it helps identify conflicts between the goals of the pilot and the goals of the automation. The remainder of this paper describes, the theory behind the AMgr, the broader motivation for it, its architecture and operation, and our plans for its evaluation.

Agenda Management

Agenda Management is defined in terms of actors, goals, functions, and resources. An **actor** is an entity (e.g., a human pilot or an autopilot) that can control or change the state of the aircraft and/or its subsystems. A **goal** is a representation (mental, electronic, or even mechanical) of an actor's intent to change the state of the aircraft or one of its subsystems in some significant way, or to maintain or keep the aircraft or one of its subsystems in some state. A **function** is an activity performed by an actor to achieve a goal. Functions performed by human actors are called tasks. Actors use **resources** to perform functions. Human actor resources include eyes, hands, memory, and attention; machine actor resources include input and output channels, memory, and processor cycles. Other machine resources include flight controls, electronic flight instrument system displays, and radios. In general, several goals might exist at any time, so several functions must be performed concurrently to achieve them. Actors must be assigned to perform those functions and resources must be allocated to enable them.

An **agenda** is a set of goals to be achieved and a set of functions to achieve those goals. **Agenda Management** (AMgt) is a high-level flightdeck function performed cooperatively by flightdeck actors which involves two sub-functions. **Goal management** is the process of recognizing or inferring the goals of all flightdeck actors, canceling goals that have been achieved or are no longer relevant, identifying and resolving conflicts between goals, and prioritizing goals consistently with safe and effective aircraft operation. **Function management** is the process of initiating functions to achieve goals, assigning actors to perform functions, assessing the status of each function (whether or not it is being performed satisfactorily and on time), prioritizing those functions based on goal priority

and function status, and allocating resources to be used to perform functions based on function priority. We consider the scope of AMgt to coincide with a subset of crew resource management (CRM).

At any point in time, **AMgt performance** is satisfactory if and only if there are no goal conflicts; all goals and functions are properly prioritized; and *either* performance of all functions is satisfactory, *or* if that is not possible, actors are actively engaged in bringing the highest priority unsatisfactory functions up to a satisfactory level of performance. In an earlier study that considered only the management of functions performed by human actors (that is, task management) we found strong evidence of function prioritization errors in 24 (7%) of 324 aircraft accidents investigated by the National Transportation Safety Board and 133 (28%) of 470 aircraft incidents reported to the Aviation Safety Reporting System (Chou et al, 1996). One recent and catastrophic instance of human actor vs. machine actor goal conflicts was the Nagoya, Japan A300 accident (Aircraft Accident Investigation Commission, 1996). From these preliminary findings we have concluded the failure to perform AMgt satisfactorily is a significant factor in flight safety. This conclusion led to AMgr development.

AgendaManager Architecture

The AMgr is implemented in Smalltalk, an object-oriented programming language. Major AMgr objects include System Agents, Actor Agents, Goal Agents, Function Agents, an Agenda Agent, and an Agenda Manager Interface. Each Agent is a simple knowledge-based object representing the corresponding elements of the cockpit environment. As a representative of such an element, the Agent's purpose is to maintain timely information about it and to perform processing that will facilitate AMgt. An Agent's declarative knowledge is represented using instance variables. Its procedural knowledge is represented using Smalltalk methods.

System Agents (SAs, e.g., the Aircraft Agent, Engine Agents) help the pilot maintain situational awareness by representing a system in the simulated environment and making state information about that system available to other Agents. **Actor Agents** (AAs, e.g., the Flightcrew Agent, the Autoflight Agent) recognize actors' goals, implicitly and explicitly, and make them known to the rest of the AMgr. The **Flightcrew** (or pilot) **Agent** is connected to a Verbex speech recognition system which allows the pilot to declare his/her intents explicitly by short vocal utterances, usually air traffic control (ATC) clearance acknowledgements. **Goal Agents** (GAs, e.g., a 'descend to 9,000 ft' Goal Agent) represent actors' goals, checking for conflicts with other goals and recognizing when goals are achieved. **Function Agents** (FAs, e.g., a 'descend to 9,000 ft' Function Agent) monitor whether the goals are being achieved in a satisfactory and timely manner. The single **Agenda Agent** is the executive Agent which coordinates the activities of all other Agents by maintaining a collection of Goal/Function Agents, initiating goal conflict assessments, and prioritizing the Agents.

Operation

As the simulator runs, AMgr System Agents maintain a situation model of the simulated aircraft and its environment. Actor Agents monitor real or simulated actors, detect or infer goals, and create instances of Goal and Function Agents. Goal Agents look for conflicts with each other and monitor the situation model to see if their goals are achieved. Function Agents monitor the progress -- if any -- made in achieving their associated goals. The Agenda Agent prioritizes Goal and Function Agents and keeps track of goal conflicts. The AMgr display presents this Agenda information to the pilot to facilitate AMgt.

AgendaManager Display

For each goal/function, the AMgr displays a short verb phrase, such as 'descend to 9,000 ft', and a brief function status message, as determined by the Function Agent. If the function is being performed satisfactorily, the text is shown in white. If not, color coding follows that of EICAS, where possible, and where AMgr functionality extends beyond that of EICAS (as in monitoring non-system-management related functions) attempts were made to be consistent with EICAS philosophy. Table 1 compares AMgr messages with corresponding EICAS messages. Gray cells represent lacking functionality. Black cells represent impossible or don't-care conditions. Though the AMgr display is still in development, Figure 2 shows the current version with some representative messages.

AgendaManager Evaluation

At the time of writing the AMgr is in final development and evaluation. Line pilots will fly the part-task simulator with and without the AMgr in a balanced experimental design. We will compare AMgt performance by measuring the time required to detect and resolve goal conflicts and by recording the proportion of time that all functions are being performed satisfactorily

Acknowledgements

References

Airbus Industrie (1989). *A320 Flight Crew Operating Manual, Systems Description*. Blagnac Cedex, France: Airbus Industrie.

Aircraft Accident Investigation Commission (1996). *Aircraft Accident Investigation Report, China Airlines Airbus Industrie A300B4-622R, B1816, Nagoya Airport, April 26, 1994*, Tokyo: Japan Ministry of Transport.

Berson, B.L., Po-Chedley, D.A., Boucek, G.P., Hanson, D.C., Leffler, M.F., and Wasson, R.L. (1981). *Aircraft Alerting Systems Standardization Study, Volume II: Aircraft Alerting System Design Guidelines*, FAA Report No. DOT/FAA/RD-81/38, II. Washington, DC: Federal Aviation Administration.

Boeing (1988). *Model 757 Operations Manual, Volume 1*. Seattle, WA: Boeing Commercial Airplanes.

Boeing (1989). *Model 747-400 Operations Manual, Volume 2*. Seattle, WA: Boeing Commercial Airplanes.

Boeing (1994). *B777-200 Operations Manual, Volume 2*. Seattle, WA: Boeing Commercial Airplanes.

Boucek, G.P., Veitengruber, J.E., and Smith, W.D. (1977). *Aircraft Alerting Systems Criteria Study, Volume II: Human Factors Guidelines and Aircraft Alerting Systems*, FAA Report FAA-RD-76-222, II. Washington, DC: Federal Aviation Administration.

Chou, C.D., Madhavan, D. and Funk, K. (1996). Studies of Cockpit Task Management Errors, *The International Journal of Aviation Psychology, 6*(4), pp. 307-320.

Cooper, G.E. (1977). *A Survey of the Status of and Philosophies Relating to Cockpit Warning Systems*. Report No. NASA CR-152071, Moffett Field, CA: NASA Ames Research Center.

McDonnell Douglas (1991). *MD11 Flight Crew Operating Manual*, Volume III, Systems Description. Long Beach, CA: McDonnell Douglas, Douglas Aircraft.

Morton, P.M. (1983). The 757/767 Engine Indicating and Crew Alerting System, Its History, Philosophy, and Future. Presentation to The Royal Aeronautical Society - London Airport Branch and The Society of Licensed Aircraft Engineers and Technologists, September 16, 1982.

Parks, D.L., Hayashi, M.M., and Fries, J.R. (1973). *Development of an Independent Altitude Monitor Concept*, Final Report No. FAA-RD-73-168. Washington, DC: Federal Aviation Administration.

Wiener, E.L. (1989). *Human factors of advanced technology (Glass Cockpit) transport aircraft* (NASA Contractor Report 177528). Moffett Field, CA: NASA-Ames Research Center.
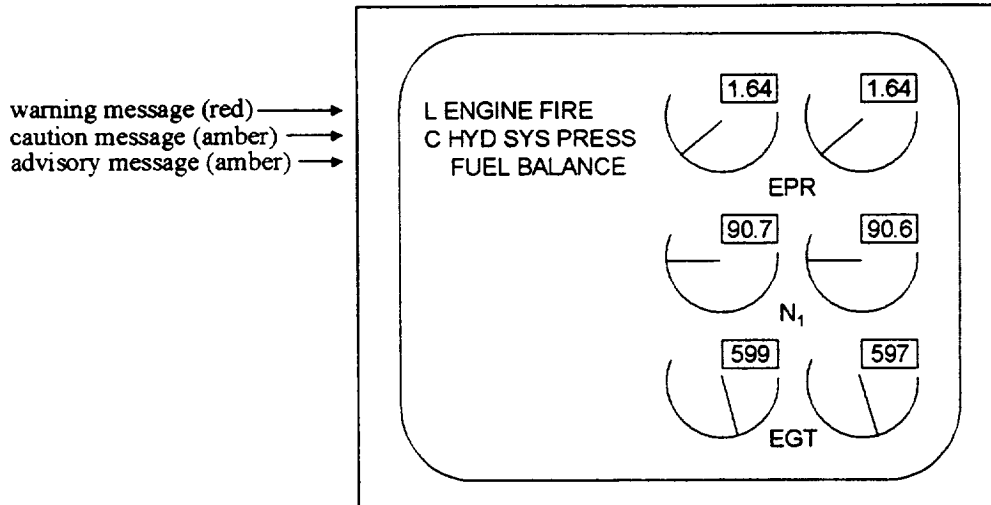
warning message (red) ——→
caution message (amber) ——→
advisory message (amber) ——→

L ENGINE FIRE
C HYD SYS PRESS
FUEL BALANCE

Figure 1. EICAS (simplified) with representative messages.

Table 1. A comparison of representative EICAS and AgendaManager messages.

EICAS message	EICAS level	EICAS color	AMgr message	AMgr priority	AMgr color function satisfactory	AMgr color function unsat.
L ENGINE FIRE	warning	red	extinguish L ENGINE FIRE	5		red
			descend to 9000 ft	4	white	amber
C HYD SYS PRESS	caution	amber	restore C HYD SYS PRESS	2		amber
FUEL BALANCE	advisory	amber	correct FUEL BALANCE	1	white	amber
			descend to 9000 ft (pilot) *vs.* descend to 8000 ft (autoflight)	goal conflict		amber

goal conflict (amber) ——→

priority 5 message (red) ——→
priority 2 message (amber) ——→

priority 4 message (green) ——→
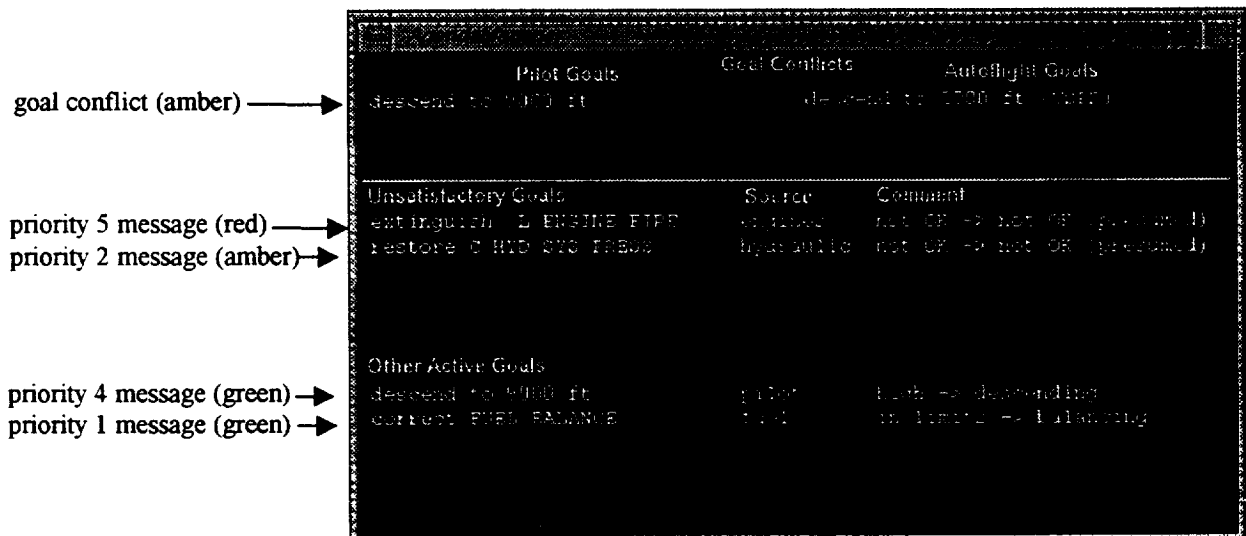priority 1 message (green) ——→

Figure 2. Current version of the AgendaManager display with representative messages.

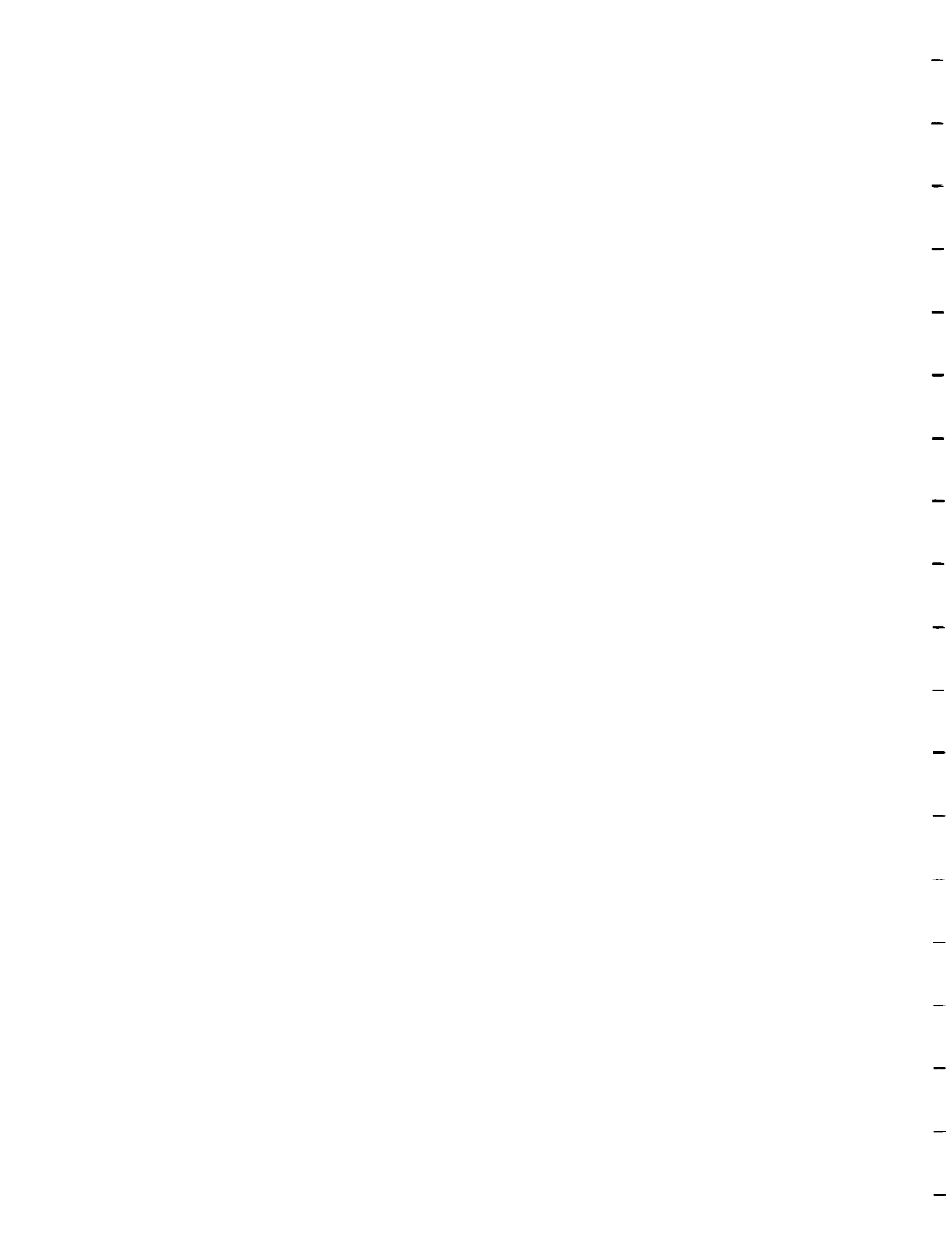# Recognizing Pilot Goals to Facilitate Agenda Management

Woo Chang Cha
Dept. of Industrial and Manufacturing Engineering
118 Covell Hall
Oregon State University
Corvallis, OR 97331-2407

541-737-5240
chaw@hfe.ie.orst.edu

Ken Funk
Dept. of Industrial and Manufacturing Engineering
118 Covell Hall
Oregon State University
Corvallis, OR 97331-2407

541-737-2357
funkk@engr.orst.edu

RECOGNIZING PILOT GOALS TO FACILITATE AGENDA MANAGEMENT

Woo Chang Cha and Ken Funk
Oregon State University
Corvallis, Oregon
USA

INTRODUCTION

In modern aircraft, the human pilots are no longer the only actors that control the aircraft and its systems. Machine actors, such as the autopilot and flight management system, also play an active role in control. In fact, several recent accidents occurred due to goal conflicts between human and machine actors. To prevent the occurrence of these and other activity management problems, a computational aid called the AgendaManager (AMgr) is being developed. The AMgr, which operates in a part-task simulator environment, attempts to facilitate the management of goals the actors are trying to accomplish and the functions being performed to accomplish them.

To provide accurate knowledge of pilot goals for the AMgr, a Goal Communication Method (GCM) was developed. The embedded GCM recognizes explicit and/or implicit pilot goals and declares them to the AMgr. This paper presents the development, architecture, operation, and evaluation of the GCM.

BACKGROUND

On an automated flightdeck the pilot (hereinafter, the human actor) must be able to monitor the automated systems (hereinafter, the machine actors) as the machine actor must be able to monitor the human actor; also, each of the two elements must be knowledgeable about the other's intentions or goals. Several intelligent procedural aids, such as the Pilot's Associate (PA) and the Cockpit Task Management System (CTMS), have been developed that utilize this cross monitoring function (Rouse et al. 1987; Kim, 1994).

However, it is often difficult for the human actor to efficiently describe the complete set of his/her goals to the machine actor. That is, the human actor has an explanation problem with respect to the machine actor. In such a complex, dynamic domain as aviation, the human ability to explain intentions to the intelligent system is highly constrained by both time and the expressive capabilities of a non-textual interface (Hammer, 1984; Hoshstrasser, 1991). Thus, recognition of pilot goals by machine actors has become an important safety issue as the use of automation increases in modern aviation systems.

A *goal* can be defined as the actor's intentions to achieve a desired system state or system behavior. Goal communication consists of the sharing of goal-directed internal representations between pilots (human actors) and intelligent subsystems (machine actors) in overt (explicit) or covert (implicit) forms that both actors readily understand. To design a goal communication framework for the control of an avionics system, it is increasingly important and useful to distinguish between overt and covert channels of communication.

Overt goal communication allows the human actor to explicitly declare his/her goals to the machine actor via such standard communication media as the control yoke, buttons and switches, and/or voice commands. In *covert* goal communication, the pilot simply performs flight tasks and a model-based intent inferencer infers goals from the procedural actions (Geddes, 1989; Gerlach et al., 1995; Rubin et al., 1988). By way of caution, it should be noted that whereas covert goal communication imposes little or no additional workload upon humans within the cockpit environment, it is subject to error due to the limits of the ability of the machine actor to correctly interpret pilot actions. And though a chance of misunderstanding poses only a slight risk in experimental laboratory studies, that slight chance may have serious effects upon aviation safety in more realistic environments. The

objective of this research was to integrate overt and covert means of goal communication to combine the reliability of the former with the low workload demands of the latter.

METHOD

The integrated overt/covert GCM was developed, implemented, and evaluated in a real-time, part-task flight simulation environment.

Flight Simulator

The simulator consisted of aerodynamic and autoflight models derived from the NASA Langley Advanced Civil Transport Simulator, primary flight displays derived from the NASA Ames Advanced Concept Flight Simulator, and subsystem models and synoptic displays developed at Oregon State University. The integrated flight simulation environment, implemented on Silicon Graphics Indigo-2 UNIX-based workstations, provided a part-task simulator that modeled a two-engine turbojet transport aircraft.

The Agenda Manager

A flightdeck agenda consists of a prioritized set of goals to be achieved and a prioritized set of functions to accomplish these goals. It is the responsibility of the flightcrew to see that goals are appropriate and consistent and that functions are performed to achieve those goals.

The Agenda Manager (AMgr), which operates in the part-task simulator environment, is a computational aid developed to facilitate management of the flightdeck agenda (Funk and Braune, 1997). The function of the AMgr is to recognize actor goals, identify goal conflicts, and monitor the progress of functions being performed to achieve the goals. The AMgr is implemented in Smalltalk, an object-oriented programming language, and runs on a Silicon Graphics Indigo-2.

The Goal Communication Method

While it is straightforward for the AMgr to recognize machine actor goals by simply noting modes and target values, recognizing human pilot goals is not so simple. The Goal Communication Method (GCM) was developed for this purpose. The GCM is embedded in the AMgr for the recognition, inferencing, updating, and monitoring of pilot goals. It uses both overt (explicit) and covert (implicit) methods of goal communication.

Overt (Explicit) Goal Communication

To declare pilot goals overtly or explicitly, a verbal modality was employed using an existing Automatic Speech Recognition system (ASR). Using this method, the subject pilots called out their goals via microphone. The overt GCM framework consisted of two main parts. The first part recognized the goals using the ASR system and the second part declared the recognized goals to the AMgr.

While a pilot is performing flightdeck operations, he or she communicates with an Air Traffic Control (ATC) controller, readily facilitating the detection of pilot goals. Since it is a legal requirement that the pilot read back all ATC clearances, pilot goals concerning the control of the aircraft's heading speed, and altitude can be recognized by monitoring these clearance acknowledgments. For example, if ATC issues the clearance "OSU 037, climb to 9000," the pilot acknowledges the clearance with a response "Roger, climb to 9000, OSU 037," and an ASR system could recognize the pilot's utterance and declare a "climb to 9000 ft" goal to the AMgr.

The ASR system used for this research was a Verbex VAT31 installed in an IBM PC compatible personal computer. The VAT31 has a 40 MHz Digital Signal Processor (DSP) running under DOS and continuous and speaker-dependent capabilities. The encoded form of verbally declared goals was sent through an RS232 serial port to the computer running the AMgr, in which the goals were parsed, declared, and maintained.

Accuracy in the recognition of pilot goals is very important. Although accuracy depends to a considerable degree upon current ASR technology, careful human factors engineering of several system design aspects helped to increase recognition accuracy; for example, vocabulary selection, user and recognizer training, visual and audio feedback, and means for correcting misrecognition (Cha, 1996).

Covert (Implicit) Goal Communication Method

While pilot goals were recognized via overt means when communicating with the ATC controller, they were also implicitly inferred from operational and/or other factors, such as the pilot actions of moving the control stick. The method for inferring goals is called covert goal communication. The covert method was implemented to avoid the workload associated with overt goal communication. To build dynamic representations of current pilot goals, the inference logic for the hypothesized current pilot intentions was based upon four components; 1) pilot actions using sensed input (e.g., throttle, stick, landing gear control), 2) aircraft state information, 3) cockpit procedures, and 4) overtly declared goals.

With knowledge of the four components, a script was constructed as a data-driven knowledge source. The script consisted of a Smalltalk representation of loosely ordered sets of pilot actions to carry out a particular goal. Given the current state of the above component variables and the current flight phase, GCM tried to interpret pilot actions based upon script-based reasoning processes. If the action could be explained by a script, the corresponding active goal was recognized and declared by the intent inferencer, which represented a process model using a blackboard problem-solving method. The knowledge source in this blackboard framework consisted of a rule-based representation of goals and corresponding scripts for the part-task simulation domain. If the actions were not predicted by a script, then the GCM asked the pilot to declare his or her goal explicitly using overt GCM.

Evaluation of the GCM

An evaluation of the GCM method of communicating pilot goals was conducted to ensure that the system correctly recognized the intentions of the human actor. In other words, the evaluation provided a measure of how well the GCM recognized pilot goals or intentions and how the GCM affected pilot performance. In a laboratory experiment using human subjects, this evaluation process demonstrated GCM effectiveness in terms of accuracy, speed, user satisfaction, and workload for the recognition of pilot goals within a simplified version of the AMgr.

Subjects The GCM was evaluated by 10 licensed general aviation pilots. Although most did not have commercial licenses and were not initially familiar with the electronic displays used in the simulator, all had some instrument flying knowledge and experience in controlling and monitoring aircraft altitude, speed, and heading. All of the subjects also had experience communicating with ATC.

Procedures To measure GCM effectiveness in terms of accuracy and workload, subjects were required to fly a simulated Eugene-to-Portland, Oregon scenario which involved declaring and achieving altitude, heading and speed goals manually. That is, the autoflight system was not used. Using the same scenario with the same conditions, one experiment was performed running with the GCM and a second without the GCM. The subject pilots called out their goals explicitly using a headset microphone. Speech patterns were collected from the subjects concurrently, as they verbalized their intentions, actions, and problem-solving activities while operating the flight simulator. While they were flying, subjects were instructed to read back ATC commands immediately after they were heard. If they failed to declare their goals verbally, they were asked to repeat their goals until the overt GCM recognized them. The successfully declared goals were displayed on the AMgr displays. The subjects also removed their goals verbally whenever this was required.

The subject goals were also declared and recognized via covert GCM, which employed the intent-inferencing mechanism based on aircraft states, subject control actions, and verbally-declared active goal as described above. Whenever the subjects took actions using thrust levers or control buttons and levers, the GCM inferred, interpreted and displayed the goals. GCM compared the subject's actions with the current active script. If the actions matched the script, the actions were explained and the corresponding goal was inferred. Whenever the subjects were aurally alerted by the GCM that their actions could not be understood, they were asked to remove the ambiguity by taking

a corrective action. If the GCM understood the corrective action, the ambiguity was resolved. If the covert GCM still failed to recognize the goal correctly, subjects were required to declare the goal verbally using overt GCM.

To measure the subject's perceived workload, the NASA-TLX multi-dimensional subjective measure was used. To facilitate accurate and objective experimental analysis, the entire flight simulation was videotaped.

RESULTS

Recognition Accuracy

GCM accuracy was measured statistically using confidence-interval estimation to determine accuracy. With the assumption of normality and a random sample of size 8 for recognition accuracy, we can say with a level of confidence of 95% that at least 87% of the explicitly declared goals after the first utterance and 99% of the implicitly declared goals were successfully recognized. Similarly, at least 93% recognition accuracy was obtained by the integrated method of covert and overt GCM. When overtly declared goals were not recognized after the first utterance, recognition accuracy after the second (corrective) utterance was 99%.

Comparison of workload

The objective of measuring workload was to know if any additional workload was imposed on subjects using GCM. It was assumed that the differences of n = 8 paired observations were normally and independently distributed random variables with mean μ_D and variance σ_D^2. The null hypothesis was that there was no additional workload when subjects used GCM. From the results shown in Table 1, the null hypothesis cannot be rejected. Therefore, it may be safely concluded that no extra workload was imposed by GCM.

Table 1 Workload comparison

legs	takeoff & climb			cruise & descend			descend & approach		
	w/ GCM	w/o GCM	difference	w/ GCM	w/o GCM	difference	w/ GCM	w/o GCM	difference
mean	3.8	3.0	0.9	1.3	1.2	0.1	4.8	4.4	0.4
variance	1.36	1.34	1.80	0.59	0.54	0.44	2.85	5.67	3.20
t_0			1.791			0.588			0.633
$t_{.05,7}$			1.895			1.895			1.895

Comparison of pilot flight control performance

The objective of measuring pilot performance in controlling flight was to know whether GCM interfered with pilot performance in controlling flight. Table 2 compares the data collected with and without GCM as a percentage of satisfactory performance. With the assumption of normality, the null hypothesis that there was no difference between performance in controlling speed, altitude, and heading with or without the GCM could not be rejected. Therefore, it may be concluded that the use of GCM did not significantly affect pilot flight control performance during the simulation.

Table 2 Flight control performance comparison chart

	speed goal			altitude goal			heading goal		
	w/ GCM	w/o GCM	diff	w/ GCM	w/o GCM	diff	w/ GCM	w/o GCM	diff
mean	68%	64%	4%	43%	43%	0%	51%	48%	3%
variance	0%	1%	1%	0%	0%	0%	2%	1%	1%
t_0			1.287			0.045			1.219
$t_{.025,7}$			2.365			2.365			2.365

DISCUSSION

Overall, the laboratory experiments conducted for the present study demonstrated the ability of the GCM to successfully recognize overt and covert goals. Specifically, the overt and covert integrated method achieved at least 93% accuracy while the overt GCM alone obtained at least 87% accuracy after the first utterance and 99% accuracy after the second (corrective) utterance. It was also indicated that the GCM neither imposed extra workload on the subjects, nor affected subjects' flight control performance.

However, this is not to say that the GCM would not face potential limitations when applied to real flight systems. The potential problems and limitations of the GCM used for this study are related to limitations in ASR technology and in intent inferencing.

Limitations To ASR Technology

Over the past two decades advances in ASR technology have contributed to a technology that has potential for aviation domains exhibiting mentally, physically and psychologically stressful environments. But, as seen from the experimental results, approximately 9% of GCM overt goal declarations were incorrect after the first utterance. This level of accuracy is not sufficient for real world applications. Nevertheless, several investigations have successfully used ASR systems for the recognition of overtly declared pilots goals in real cockpit environments, leading to the overall conclusion that most overt goal recognition errors could be removed by repeating declarations of unrecognized goals or by the application of updated ASR technologies (Williamson, 1996; Gerlach et al., 1995). In fact, the experimental results from the present study demonstrated that the second utterances for failed goal recognition achieved close to 100% accuracy. Thus, if we accept the costs of second trials or of the inclusion of advanced technologies, the GCM can be considered to be an accurate means of goal communication.

Limitations to Intent Inferencing

To resolve the workload associated with overt communications, the present study employed a model-based inferencer to infer pilot goals. Although the experimental results showed almost perfect recognition accuracy of the covert goals, the accuracy of the covert GCM probably resulted in large part from the fact that the inferencing was done in a highly simplified environment and was based on limited actions, simple scripts and rules, and simple scenarios. The effective use of intent inferencing in a more realistic environment would require a more robust intent inferencing mechanism such as the Georgia Tech crew-activity tracking system (GT-CATS) (Callantine and Mitchell, 1994). To infer the flightcrew goals, GT-CATS decomposes operator function into automatic control modes, which can be used to perform the functions. Each mode in turn decomposes into the tasks, subtasks, and actions required to use it, depending on the situation.

Conclusion

Insofar as it was demonstrated that the GCM developed for the present study has the capacity to recognize pilot goals with a high degree of accuracy and with little or no increase in workload, we conclude that GCM is suitable for use in the AgendaManager, at least for development purposes. To the extent that the use of the AMgr is restricted, for the time being at least, to laboratory or training environments, GCM should be a suitable 'front end' to correctly recognize pilot goals. Future implementations of the AMgr in real aircraft will require better automatic speech recognition systems and more robust intent inferencing mechanisms.

ACKNOWLEDGMENT

REFERENCES

Cha, W. (1996). *Communicating Pilot Goals to an Intelligent Cockpit Aiding System*. Ph.D. Thesis. Corvallis, OR: Dept. of Industrial and Manufacturing Engineering, Oregon State University.

Funk, K. & Braune, R.J. (1997). Expanding the functionality of existing airframe system monitors: the AgendaManager. *Proceedings of the Ninth International Symposium on Aviation Psychology*, April 27 - May 1, Columbus, Ohio.

Geddes, N.D. (1989). *Understanding Human Operator's Intentions in Complex Systems*. Ph.D. Thesis. Atlanta, GA: Georgia Institute of Technology.

Gerlach, M., Prevot, T., Ruckdeschel, W., Wittig, T., & Onken, R. (1995). Evaluation of intelligent on-board pilot assistance in in-flight field trials. *IFAC Man-Machine Systems '95*, June 27-29, Cambridge, MA.

Hammer, J. (1984). An intelligent flight management aid for procedure execution. *IEEE Transactions On Systems, Man, And Cybernetics, SMC-14*, No. 6, 885-888.

Hoshstrasser, B.H. (1991). The design of an object-oriented intent inferencer for integrated systems. *Digital Avionics Systems Conference*. October 14-17, Los Angeles.

Kim, J.N. (1994). *An Agent-Based Cockpit Task Management System: a Task-Oriented Pilot-Vehicle Interface*. Ph.D. Thesis. Corvallis, OR: Dept. of Industrial and Manufacturing Engineering, Oregon State University.

Callentine, T.J. & Mitchell, C.M. (1994). A methodology for understanding how operators select and use automation to control complex dynamic systems. *Proceedings of the 1994 International Conference on Systems, Man, and Cybernetics*. October , San Antonio. TX.

Rouse, W.B., Geddes. N.D., & Curry, R.E. (1987). An architecture for intelligent interfaces: outline of an approach to supporting operators of complex systems. *Human Computer Interaction, 3*, 87-122.

Rubin, K.S., Jones, P.M. & Mitchell, C.M. (1988). OFMspert: inference of operator intentions in supervisory control using a blackboard architecture. *IEEE Transaction On Systems, Man, And Cybernetics, SMC-18*, No. 4, 618-637.

Williamson, D.T., Barry, T.P. & Liggett, K.K. (1996). Flight test results of ITT VRS-1290 in NASA OV-10. *Proceedings of American Voice Input/Output Society*. September 10-12, San Jose, CA.

Communicating Pilot Goals to an Intelligent Cockpit Aiding System

Woo Chang Cha and Ken Funk
Oregon State University
Corvallis, Oregon
USA

INTRODUCTION

In modern aircraft, the human pilots are no longer the only actors that control the aircraft and its systems. Machines, such as the autopilot and flight management system, also play an active role in control. In fact, several recent accidents occurred due to goal conflicts between human and machines. To facilitate the coordination of these actors, a computational aid called the AgendaManager (AMgr) is being developed. The AMgr, which operates in a part-task simulator environment, attempts to facilitate the management of goals the actors are trying to accomplish and the functions being performed to accomplish them. To provide accurate knowledge of pilot goals for the AMgr, a goal communication method (GCM) was developed. The embedded GCM recognizes explicit and/or implicit pilot goals and declares them to the AMgr. This paper presents the development, architecture, operation, and evaluation of the GCM.

BACKGROUND

The AgendaManager

A *goal* is a desired aircraft or aircraft subsystem state or behavior. For example, 'climb to 9000 feet' or 'restore fuel pressure to right engine' are goals. A *function* is an activity performed to achieve a goal. Goals are declared and functions are performed by *actors*. Human actors are pilots. Machines include autoflight and flight management systems. An *agenda* is a set of goals and functions. *Agenda Management* (AMgt) is a high level function performed by the flightcrew that involves

1. assessing the goals of all actors, removing those that are achieved, inappropriate, or inconsistent;
2. assessing the functions being performed to achieve those goals to see that satisfactory progress is being made towards achieving the goals;
3. prioritizing the functions, based on the importance and urgency of the goals and the status of the functions; and
4. allocating actor attention to the functions in order of assessed priority.

Ideally, AMgt is performed continuously by the flightcrew, so that all appropriate goals are achieved and that the higher priority functions are performed before the lower priority ones.

In fact, that does not always happen. In analyses of 324 National Transportation Safety Board aircraft accident reports and 450 Aviation Safety Reporting System aircraft incident reports, we found that improper AMgt contributed to 76 (23 %) aircraft accidents and 231 (49 %) aircraft incidents (Chou et al, 1996).

As one possible approach to dealing with this problem, we are developing an experimental, computational aid to facilitate AMgt called the AgendaManager (AMgr). The AMgr operates in a part-task simulator environment, which is described below. It is an agent-based system made up of a collection of software modules called agents. Each agent represents some entity in the simulated flightdeck environment.

System agents represent aircraft systems, such as engines and the fuel system. Each system agent maintains current state information on its system, such as engine speed or fuel pressure, and detects system faults, such as engine fires or fuel pressure drops.

Goal agents represent actor goals. Each goal agent is capable of recognizing the conditions necessary for goal

achievement. Additionally, goal agents recognize goal conflicts, such as would occur when the pilot's goal was to climb to 9,000 ft but the autoflight system's target altitude was inadvertently set to 8,000 ft.

Function agents represent the functions being performed to achieve the goals. A function agent records the status of the function and assesses function performance. For example, a 'climb to 9,000 ft' function agent knows that its function has a high priority (because altitude control is critical to flight safety) and can determine if the aircraft's altitude is changing towards the 9,000 ft target value.

Actor agents are a special kind of system agent representing actors. The autoflight agent keeps track of the autoflight system's goals by noting its modes and target values and instantiating goal agents. The flightcrew agent keeps track of the simulator pilot's goals in a manner described below.

The AMgr interface consists of a display that informs the pilot of goal conflicts and the status of each function, thereby facilitating AMgt. As the pilot "flies" the simulator, either manually or by using the autoflight system, system agents monitor aircraft and aircraft system state, and when faults are detected, instantiate goal agents for goals to correct them. Actor agents recognize actor goals to control the aircraft and instantiate corresponding goal agents. Goal agents check for goal conflicts and inform the pilot of any via the AMgr display. Function agents continually monitor the progress of functions to achieve the goals and inform the pilot if any are not being performed satisfactorily. The pilot is thus informed of the state of the simulated flightdeck environment and AMgt is facilitated.

Goal Communication

But this process can work only if the pilot can make his/her goals known to the AMgr. This is a special case of the human-machine goal communication problem. In fact, it is often difficult for the human actor to efficiently describe the complete set of his/her goals to a machine such as the AMgr. That is, the human actor has an explanation problem with respect to the machine. In such a complex, dynamic domain as aviation, human ability to explain intentions to the intelligent system is highly constrained by both time and the expressive capabilities of a non-textual interface (Hammer, 1984; Hoshstrasser, 1991). Thus, recognition of pilot goals by machines has become an important safety issue as the use of automation increases in modern aviation systems.

Goal communication consists of the sharing of goal representations between human actors and intelligent machines in overt (explicit) or covert (implicit) forms that both the human and the machine readily understand. To design a goal communication framework for the control of an avionics system, it is increasingly important and useful to distinguish between overt and covert channels of communication.

Overt Goal Communication

Overt goal communication is an activity which allows the human actor to explicitly declare goals to a machine, such as the AMgr. One set of general alternatives consists of such standard communication media as the control yoke, buttons and switches, a keyboard, a touch panel, a mouse, and/or voice commands. For example, the human actor communicates a goal to the autopilot (A/P) subsystem via the mode control panel (MCP), which consists of several interrelated knobs and buttons. If the human actor wants to engage the autopilot, then the goal is stated explicitly by simply activating the A/P switch on the MCP. Or, the human actor may tell the flight management system (FMS) by keystrokes on the Control Display Unit (CDU) to follow a certain flight path, and the FMS responds by informing the human actor of the estimated time of arrival and rate of fuel consumption. Finding these estimates acceptable, the human actor explicitly instructs the FMS to implement the plan via the CDU. Standard input devices such as buttons and keyboard, used as overt communication media, often fail to recognize pilot goals directly and accurately because human pilots are fallible in their operation of buttons and switches, and because the pilots may experience additional cognitive loading to perform the operations.

All activities that declare a pilot's goals explicitly are considered to be explicit goal communications, even should such communications imply covert communications. For example, if the pilot should push the flight level change switch on the MCP to the on position, the activity itself is explicit goal communication, since the pilot has

explicitly declared the goal of changing the altitude. At the same time, such a goal would automatically imply the holding of current heading and to trigger vertical speed modes. Goals for the heading hold and vertical speed modes will be implicitly declared from the implicit goal communication method.

Although the technology for speech interaction between humans and machines is by no means perfect, Automatic Speech Recognition (ASR) technology has received increased attention as an input means for direct and accurate overt goal communication. And despite the fact that current ASR technology has focused heavily on telecommunication applications such as voice activated telephone services, ASR is considered to be a promising method to declare pilot goals in a wide range of airborne environments, from helicopters and military jets (Mountford and North, 1980; Reed, 1985; Williamson et al., 1996) to civil aircraft (Starr, 1993). The application domain of flying an airplane is recognized as being potentially challenging to the use of ASR, since it exhibits some attributes that characterize adverse environments for ASR, such as high noise levels, high acceleration forces, and extreme levels of workload and stress (Williamson et al., 1996; Baber and Noyes, 1996). Nevertheless, ASR has been increasingly explored in the aviation domain not only because of its potential to reduce pilot workload; ASR permits "eyes-and hands-free" interaction with flight control systems and allows pilots to maintain head-up flight with "hands on throttle and stick" control. The potential exists also because of the fact that pilots are consistently communicating their goals verbally with air-traffic controllers and other flightcrew members, and because ASR technology is advancing rapidly.

Covert Goal Communication

The control actions of the pilot as he/she controls the aircraft by means of yoke, rudder pedals, throttles, and other controls implicitly carry within them information about the pilot's goals. Such goal information is available to and could be interpreted by an intelligent machine, such as the AMgr. This form of goal communication is covert in the sense that the human need not be conscious of the information transformation process.

There are two primary reasons for trying to use covert goal communication. The first reason is to avoid the workload associated with overt communication. For example, if the machine could be enabled to covertly assess the human actor's intentions, then the human would not be distracted from other activities for the purpose of supplying this information. The second motive for the use of covert goal communication is based upon the possibility that, at certain times or in certain situations, it will not be possible to communicate goals overtly due to the fact that hands and voice are fully occupied with other, safety critical activities.

To communicate covertly or implicitly with an intelligent aid in a highly dynamic system, the human actor simply performs procedural steps and a model-based intent inferencer infers goals from the procedural actions (Gerlach et al., 1995; Onken and Prevot, 1994; Geddes, 1985, 1989; Mitchell, 1987; Rubin et al., 1988). In other words, covert communication models are embedded within the intent inferencer and compared with human actions in an attempt to infer what the human's goals are.

Integration of Overt and Covert Goal Communication

Whereas covert goal communication imposes little or no additional workload upon the human actor, control actions can be ambiguous with respect to pilot intent, and misunderstanding of pilot goals by an intent inferencer is a real possibility. And though a misunderstanding poses little risk in experimental laboratory studies, it could be catastrophic in more realistic environments. On the other hand, overt goal communication by voice or manual means imposes additional workload and may interfere with safety critical activities.

A possible solution to this dilemma is the integration of overt and covert goal communication. Hopefully, such an integrated method would offer the reliability of overt communication and the low workload requirements of covert communication.

RESEARCH OBJECTIVES

The principal goal of this research was to develop an integrated method of overt and covert (explicit and implicit) goal communication, to be embedded within the AMgr to facilitate AMgt performance. The objectives of this experimental investigation were to

1. develop a goal communication method (GCM) to recognize pilot goals based upon the integration of implicit (covert) as well as explicit (overt) modes of communication and
2. evaluate the methodology in the context of a real-time flight simulation environment with respect to
   - GCM accuracy,
   - GCM speed,
   - user satisfaction with GCM,
   - workload imposed by GCM, and
   - pilot flight control performance while using GCM.

METHOD

The integrated overt/covert GCM was developed, implemented, and evaluated in a real-time, part-task flight simulation environment.

Flight Simulator

The simulator consisted of aerodynamic and autoflight models derived from the NASA Langley Advanced Civil Transport Simulator, primary flight displays derived from the NASA Ames Advanced Concept Flight Simulator, and subsystem models and synoptic displays developed at Oregon State University. The integrated flight simulation environment, implemented on Silicon Graphics Indigo-2 UNIX-based workstations, provided a part-task simulator that modeled a two-engine turbojet transport aircraft.

The Goal Communication Method

While it is straightforward for the AMgr to recognize machine goals by simply noting modes and target values, recognizing human pilot goals is not so simple. The Goal Communication Method (GCM) was developed for this purpose. The GCM is embedded in the AMgr for the recognition, inferencing, updating, and monitoring of pilot goals. It uses both overt (explicit) and covert (implicit) methods of goal communication.

Overt (Explicit) Goal Communication

To declare pilot goals overtly or explicitly, the verbal modality was employed using a commercial automatic speech recognition system (ASR). Using the ASR, the subject pilots called out their goals via microphone. The overt GCM framework consisted of two main parts. One was to recognize the goals from the ASR system process and the second was to declare the recognized goals to the AMgr.

While a pilot is performing flightdeck operations, he/she communicates with an air traffic control (ATC) controller, readily facilitating the detection of his/her goals. Since it is a legal requirement that the pilot read back ATC clearances, pilot goals concerning the control of the aircraft's heading speed, and altitude can be recognized by monitoring these clearance acknowledgments. For example, if ATC issues the clearance "OSU 037, climb to 9000," the pilot acknowledges the clearance with a response "Roger, climb to 9000, OSU 037," and an ASR system could recognize the pilot's utterance and declare a "climb to 9000 ft" goal to the AMgr.

The ASR system used for this research was a Verbex VAT31 installed in an IBM PC compatible personal computer. The VAT31 has a 40 MHz Digital Signal Processor (DSP) running under DOS and continuous and speaker-dependent capabilities. The Verbex grammar definition file defined vocabulary and grammar for a subset

of pilot-to-ATC controller communication. Subject voice pattern files was created using the voice recognizer training process. As utterances were made by the subjeccts, the encoded form of verbally declared goals was sent through an RS232 serial port to the computer running the AMgr, in which the goals were parsed, declared, and stored.

Accuracy in the recognition of pilot goals is very important. Although accuracy depends to a considerable degree upon current ASR technology, careful human factors engineering of several system design aspects helped to increase recognition accuracy; for example, vocabulary selection, user and recognizer training, and visual and audio feedback (Cha, 1996).

Covert (Implicit) Goal Communication Method

While pilot goals were recognized via overt means when communicating with the ATC controller, they were also implicitly inferred from operational and/or other factors, such as the pilot actions of moving the control stick. This method for recognizing goals is a form of covert goal communication. The covert method was implemented to avoid the workload associated with overt goal communication. To build dynamic representations of current pilot goals, the inference logic for the hypothesized current pilot intentions was based upon four components;

1. pilot actions using sensed input (e.g., throttle, stick, landing gear control),
2. aircraft state information,
3. flightdeck procedures, and
4. overtly declared goals.

With knowledge of the four components, for each goal a script was constructed as a data-driven knowledge source. The script consisted of a representation of a loosely ordered set of pilot actions to carry out the goal (see Table 1).

Table 1 An Example Of Active Speed Script

speedScript
. . . .
overtTargetSpeed isNil ifFalse:[inferredTargetSpeed := overtTargetSpeed].
. . . .
action = #thrustLeverUp
 ifTrue:
  [phase = #beforeTakeoff
   ifTrue:
       [inferredSpeedGoal := #maintainTakeoffSpeed.
        inferredTargetSpeed := rotateSpeed.]
      ifFalse: [inferredSpeed := #maintainSpeed].
    inferredTargetSpeed = nil ifTrue:[inferredSpeedGoal := #increaseSpeed]
  ^self].
. . . .
inferredSpeedGoal := #notUnderstoodPilotAction.
^self

Given the current state of the above component variables and flight phases, GCM tried to interpret pilot actions based upon script-based reasoning processes (see Figure 1). If the action could be explained by an active script, the corresponding active goal was recognized and declared by the intent inferencer, which represented a process model using a blackboard problem-solving method. The knowledge source in this blackboard framework consisted of a rule-based representation of goals and corresponding scripts for the part-task simulation domain. If the actions were not predicted by the active script, then the GCM would ask the pilot to ignore the covert GCM and

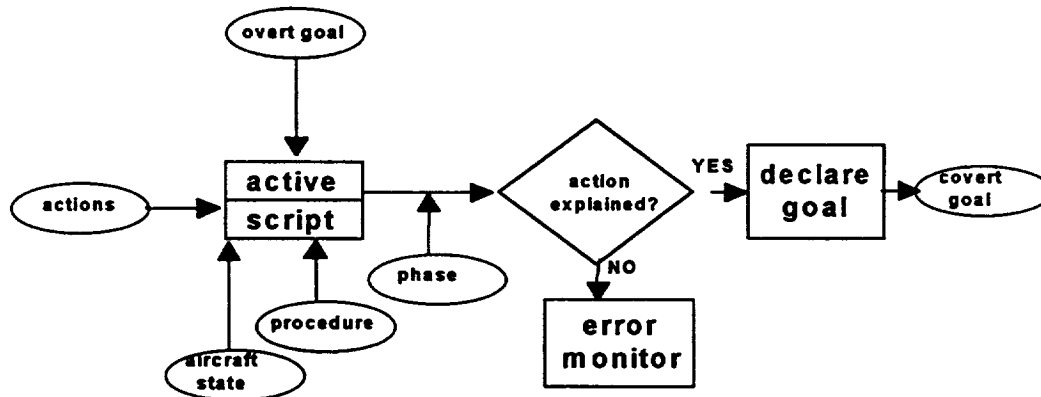declare his or her goal explicitly using overt GCM.

Figure 1 Covert GCM Process

Evaluation of the GCM

An evaluation of the GCM was conducted to ensure that the system correctly recognized the intentions of the human actor. In other words, the evaluation provided a measure of how well the GCM recognized pilot goals or intentions and how the GCM affected pilot performance. In a laboratory experiment using human subjects, this evaluation process demonstrated GCM effectiveness in terms of accuracy, speed, user satisfaction, and workload for the recognition of pilot goals within a simplified version of the AMgr.

Subjects The GCM was evaluated by 10 licensed general aviation pilots. Although most did not have commercial licenses and were not initially familiar with the electronic displays used in the simulator, all had some instrument flying knowledge and experience in controlling and monitoring aircraft altitude, speed, and heading. All of the subjects also had experience in air traffic control (ATC) communication.

Procedures To measure GCM effectiveness in terms of accuracy and workload, subjects were required to fly a simulated Eugene-to-Portland, Oregon scenario which involved declaring goals and performing tasks to control altitude, heading and speed manually. The autoflight system was not used. Using the same scenario with the same conditions, one experiment was performed running with the GCM and a second without the GCM. The subject pilots called out their goals explicitly using a headset microphone. Speech patterns were collected from the subjects concurrently, as they verbalized their intentions, actions, and problem-solving activities while operating the flight simulator. While they were flying, subjects were supposed to read back ATC commands immediately after they were heard. If they failed to declare their goals verbally, they were asked to repeat their goals until the overt GCM recognized them. The successfully declared goals were displayed on the AMgr displays. The subjects also removed their goals verbally whenever this was required.

The subject goals were also declared and recognized via covert GCM, which employed the intent-inferencing mechanism based on aircraft states, subject control actions, and verbally-declared active goal as described above. Whenever the subjects took actions using thrust levers or control buttons and levers, the GCM inferred, interpreted and displayed the goals. GCM compared the subject's actions with the current active script. If the actions matched the script, the actions were explained and the corresponding goal was inferred. Whenever the subjects were aurally alerted by the GCM that their actions could not be understood, they were asked to remove the ambiguity by taking a corrective action. If the GCM understood the corrective action, the ambiguity was resolved. If the covert GCM still failed to recognize the goal correctly, subjects were required to declare the goal verbally using overt GCM.

To measure the subject's perceived workload, the NASA-TLX (task load index) multi-dimensional subjective measure was used (Hart & Staveland, 1988). To facilitate accurate and objective experimental analysis, the entire flight simulation was videotaped.

RESULTS

Recognition Accuracy

GCM accuracy was measured statistically using confidence-interval estimation to determine accuracy. With the assumption of normality and a random sample of size 8 for recognition accuracy, we can say with a level of confidence of 95% that at least 87% of the explicitly declared goals after the first utterance, and 99% of the implicitly declared goals were successfully recognized. Similarly, at least 93% recognition accuracy was obtained by the integrated method of covert and overt GCM. When overtly declared goals were not recognized after the first utterance, recognition accuracy after the second (corrective) utterance was 99%.

Comparison of workload

The objective of measuring workload was to know if any additional workload was imposed on subjects using GCM. It was assumed that the differences of n = 8 paired observations were normally and independently distributed random variables with mean μ_D and variance σ_D^2. The null hypothesis was that there was no additional workload when subjects used GCM. From the results shown in Table 2, the null hypothesis cannot be rejected. Therefore, it may be safely concluded that no extra workload was imposed by GCM.

Table 2 Workload comparison

legs	takeoff & climb			cruise & descend			descend & approach		
	w/ GCM	w/o GCM	difference	w/ GCM	w/o GCM	difference	w/ GCM	w/o GCM	difference
mean	3.8	3.0	0.9	1.3	1.2	0.1	4.8	4.4	0.4
variance	1.36	1.34	1.80	0.59	0.54	0.44	2.85	5.67	3.20
t_0			1.791			0.588			0.633
$t_{.05,7}$			1.895			1.895			1.895

Comparison of pilot flight control performance

The objective of measuring pilot performance in controlling flight was to know whether GCM interfered with pilot performance in controlling flight. Table 3 compares the data collected with and without GCM as a percentage of satisfactory performance. With the assumption of normality, the null hypothesis that there was no difference between performance in controlling speed, altitude, and heading with or without the GCM could not be rejected. Therefore, it may be concluded that the use of GCM did not significantly affect pilot flight control performance during the simulation.

Table 3 Flight control performance comparison chart

	speed goal			altitude goal			heading goal		
	w/ GCM	w/o GCM	diff	w/ GCM	w/o GCM	diff	w/ GCM	w/o GCM	diff
mean	68%	64%	4%	43%	43%	0%	51%	48%	3%
variance	0%	1%	1%	0%	0%	0%	2%	1%	1%
t_0			1.287			0.045			1.219
$t_{.025,7}$			2.365			2.365			2.365

DISCUSSION

Overall, the laboratory experiments conducted for the present study demonstrated the ability of the GCM to successfully recognize overt and covert goals. Specifically, the overt and covert integrated method achieved at least 93% accuracy while the overt GCM alone obtained at least 87% accuracy after the first utterance and 99% accuracy after the second (corrective) utterance. It was also indicated that the GCM neither imposed extra workload on the subjects, nor affected subjects' flight control performance.

However, this is not to say that the GCM would not face potential limitations when applied to real flight systems. The potential problems and limitations of the GCM used for this study are related to limitations in ASR technology and in intent inferencing.

Limitations To ASR Technology

Over the past two decades advances in ASR technology have contributed to a technology that has potential for aviation domains exhibiting mentally, physically and psychologically stressful environments. But, as seen from the experimental results, approximately 9% of GCM overt goal declarations were incorrect after the first utterance. This level of accuracy is not sufficient for real world applications. Nevertheless, several investigations have successfully used ASR systems for the recognition of overtly declared pilots goals in real cockpit environments, leading to the overall conclusion that most overt goal recognition errors could be removed by repeating declarations of unrecognized goals or by the application of updated ASR technologies (Williamson, 1996; Gerlach et al., 1995). In fact, the experimental results from the present study demonstrated that the second utterances for failed goal recognition achieved close to 100% accuracy. Thus, if we accept the costs of second trials or of the inclusion of advanced technologies, the GCM can be considered to be an accurate means of goal communication.

Limitations to Intent Inferencing

To resolve the workload associated with overt communications, the present study employed a model-based inferencer to infer pilot goals. Although the experimental results showed almost perfect recognition accuracy of the covert goals, the accuracy of the covert GCM probably resulted in large part from the fact that the inferencing was done in a highly simplified environment and was based on limited actions, simple scripts and rules, and simple scenarios. The effective use of intent inferencing in a more realistic environment would require a more robust intent inferencing mechanism such as the Georgia Tech crew-activity tracking system (GT-CATS) (Callantine and Mitchell, 1994). To infer the flightcrew goals, GT-CATS decomposes operator function into automatic control modes, which can be used to perform the functions. Each mode in turn decomposes into the tasks, subtasks, and actions required to use it, depending on the situation.

Conclusion

Insofar as it was demonstrated that the GCM developed for the present study has the capacity to recognize pilot goals with a high degree of accuracy and with little or no increase in workload, we conclude that GCM is suitable for use in the AgendaManager, at least for development purposes. To the extent that the use of the AMgr is restricted, for the time being at least, to laboratory or training environments, GCM should be a suitable 'front end' to correctly recognize pilot goals. Future implementations of the AMgr in real aircraft will require better automatic speech recognition systems and more robust intent inferencing mechanisms.

ACKNOWLEDGMENT

REFERENCES

Baber, C. and Noyes, J. (1996). Automatic speech recognition in adverse environments. *Human Factors*, 38(1), 142-155.

Cha, W. (1996). *Communicating Pilot Goals to an Intelligent Cockpit Aiding System*. Ph.D. Thesis. Dept. of Industrial and Manufacturing Engineering, Oregon State University.

Chou, C, Madhavan, D. & Funk, K. (1996). Studies of cockpit task management errors. *The international Journal of Aviation Psychology*. Vol. 6, No. 4, 1996, pp. 307-320.

Geddes, N.D. (1985) Intent inferencing using script and plans. *Proceeding of the First Annual aerospace Applications of Artificial Intelligence Conference*, Dayton, OH. 160-172.

Geddes, N.D. (1989). *Understanding Human Operator's Intentions in Complex Systems*. Ph.D. Thesis. Georgia Institute of Technology, Atlanta, GA.

Gerlach, M., Prevot, T., Ruckdeschel, W., Wittig, T., & Onken, R. (1995). Evaluation of intelligent on-board pilot assistance in in-flight field trials. *IFAC Man-Machine Systems '95*, June 27-29, 1995, Cambridge, MA.

Hammer, J. (1984). An intelligent flight management aid for procedure execution. *IEEE Transaction On Systems, Man, And Cybernetics, SMC-14*, No. 6 885-888

Hart, S.G. & Staveland, L.E. (1988). Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In P.A. Hancock & N. Meshkati (eds.), *Human Mental Workload*. Amsterdam: N.Holland

Hoshstrasser (1991). The design of an object-oriented intent inferencer for integrated systems. *Digital Avionics Systems Conference*. Oct. 14-17, LA.

Callentine, T.J. & Mitchell, C.M. (1994). A methodology for understanding how operators select and use automation to control complex dynamic systems. *Proceedings of the 1994 International Conference on Systems, Man, and Cybernetics*, San Antonio, October.

Mitchell, C.M. (1987) GT-MSOCC: A research domain for modeling human -computer interaction and aiding decision making in supervisory control systems. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-17, 553-570.

Mountford, S.J., and North, R.A. (1980) Voice entry for reducing pilot workload. In *Proceeding of the Human Factors Society 29th Annual Meeting* (pp. 185-189), Santa Monica, CA.

Onken, R. and Prevot, T. (1994). CASSY-Cockpit assistant system for IFR operation. *ICAS 94-8.7.2 Proceedings* Vol.2, Anaheim, CA.

Reed, L. (1985). Military application of voice technology. *Speech Technology*, 2(4), 42-50.

Rubin, K.S., Jones, P.M. & Mitchell, C.M. (1988). OFMspert: inference of operator intentions in supervisory control using a blackboard architecture *IEEE Transaction On Systems, Man, And Cybernetics, SMC-18*, No. 4 618-637.

Starr, A.F. (1993). Is control by voice the right answer for the avionics environment? In C. Baber & J.M. Noyed (Eds.) *Interactive Speech Technology* (pp. 85-97). London: Taylor & Francis.

Williamson, D.T., Barry, T.P. & Liggett, K.K. (1996). Flight test results of ITT VRS-1290 in NASA OV-10. *Proceedings of American Voice Input/Output Society*, San Jose, CA, Sep 10-12.

Development and Evaluation of an Aid to Facilitate Agenda Management

**Ken Funk, Woo Chang Cha,
Robert Wilson, and Joachim Zaspel**
Department of Industrial and Manufacturing Engineering
118 Covell Hall, Oregon State University
Corvallis, OR 97331-2407, USA
541-737-2357; funkk@engr.orst.edu

Rolf Braune

7250 Old Redmond Road, #140
Redmond, WA 98052, USA
204-885-1943; 71611.1126@compuserve.com

Commercial air transportation has an admirable safety record, yet each year hundreds of lives and hundreds of millions of dollars worth of property are lost in air crashes in the United States alone. About two-thirds of these aircraft accidents are caused, in part, by pilot error. Many of these errors are errors in *performing* flightdeck (or cockpit) functions, others are errors in *managing* flightdeck goals and the functions to achieve those goals. This paper describes the development and evaluation of a prototype computational aid to facilitate the management of flightdeck goals and functions.

Background: Agenda Management

The concept of Agenda Management is an extension of a theory of Cockpit Task Management proposed by Funk [5]. Informally speaking, an agenda is a list of things to be done. So, managing a flightdeck agenda can be described informally as managing the intentions of the flightcrew and flightdeck automation and managing their activities to fulfill those intentions.

More formally, **Agenda Management** is described in terms of actors, goals, functions, and resources. An **actor** is an entity that *does* something in that it can control or change the state of the aircraft and/or its subsystems. Pilots are human actors; machine actors include autoflight and flight management systems. A **goal** is a representation (mental, electronic, or even mechanical) of an actor's intent to change the state of the aircraft or one of its subsystems in some significant way, or to maintain or keep the aircraft or one of its subsystems in some state. For example, a pilot might have a goal to descend to an altitude of 9,000 ft, a goal to maintain the current heading of 270°, and a goal to crossfeed fuel to correct a fuel system imbalance. If configured properly, the autoflight system in this example would also have a goal to descend to 9,000 ft and a goal to hold 270°. Goals come about as a result of planning and decision making in the case of human actors, and computation or human input, in the case of machine actors.

A **function** is an activity performed by an actor to achieve a goal. That activity may directly achieve the goal or it may produce sub-goals which, when achieved by performing sub-functions, satisfy the conditions of the original goal. Actors use **resources** to perform functions. Human actor resources include eyes, hands, memory, and attention; machine actor resources include input and output channels, memory, and processor cycles. Other machine resources include flight controls, electronic flight instrument system displays, and radios. In general, several goals might exist at any time, so several functions must be performed concurrently to achieve them. Actors must be assigned to perform those functions and resources must be allocated to enable them. An **agenda** then is a set of goals to be achieved and a set of functions to achieve those goals.

Agenda Management (AMgt) is a high-level flightdeck function performed cooperatively by flightdeck actors which involves two sub-functions:
1. **Goal management** is the process of
    1.1. recognizing or inferring the goals of all flightdeck actors;
    1.2. canceling goals that have been achieved or are no longer relevant;
    1.3. identifying and resolving conflicts between goals; and
    1.4. prioritizing goals consistently with safe and effective aircraft operation.
2. **Function management** is the process of
    2.1. initiating functions to achieve goals;
    2.2. assigning actors to perform functions;
    2.3. assessing the status of each function (whether or not it is being performed satisfactorily and on time);
    2.4. prioritizing those functions based on goal priority and function status; and
    2.5. allocating resources to be used to perform functions based on function priority.

At any point in time, **AMgt performance** is satisfactory if and only if:
1. there are no goal conflicts;
2. all goals and functions are properly prioritized; and
3. *either*
    3.1. performance of all functions is satisfactory, *or*

3.2. if that is not possible, actors are actively
engaged in bringing the highest priority
unsatisfactory functions up to a satisfactory
level of performance.

In an earlier study that considered only the management
of functions performed by human actors (that is, task
management [4]) we found strong evidence of function
prioritization errors in 24 (7%) of 324 aircraft accidents
investigated by the National Transportation Safety Board
and 133 (28%) of 470 aircraft incidents reported to the
Aviation Safety Reporting System. Two recent aircraft
accidents illustrate human actor vs. machine actor goal
conflicts. In 1994 in a China Airlines Airbus A300 on
approach to Nagoya, Japan, the flightcrew inadvertently
initiated an autoflight system go-around maneuver while
trying to continue the landing [2]. The goal conflict
between the flightcrew and the autoflight system caused
an out-of-trim condition that resulted in a stall and crash
which killed 264 persons. In an American Airlines
Boeing 757 on approach to Cali, Columbia in 1995, the
flightcrew accepted an air traffic control clearance direct
to a designated navigational fix [1]. They inadvertently
configured the aircraft's flight management system to fly
the airplane to a different fix. This goal conflict was not
detected in time to prevent the aircraft from crashing into
mountainous terrain, killing 159 persons.

Objectives

From these preliminary findings we have concluded that
AMgt -- and specifically the failure to perform AMgt
satisfactorily -- is a significant factor in flight safety. The
objectives of our research were to develop and evaluate
an experimental computational aid to facilitate AMgt.
We call this aid the AgendaManager.

The AgendaManager

Simulator Environment
Our part-task simulator models a generic, twin engine
transport aircraft. It is built from components developed
at the NASA Langley and NASA Ames Research centers
and in our own lab. It runs on one or two Silicon
Graphics Indigo 2 computers and provides a simplified
aerodynamic model (Langley), autoflight system
(Langley), Flight Management System (Langley),
primary flight displays (Ames), Mode Control Panel
(Ames), and system models and system synoptic displays
(OSU). The software is written in C, FORTRAN, and
Smalltalk (VisualWorks 2.5).

Analysis and Design
As a first step in designing the AgendaManager, we
developed a formal, functional model of Agenda

Management using IDEF0, a graphical modeling
methodology useful for representing and decomposing
complex activities. IDEF0 helps the analyst represent
activities, inputs and outputs to and from those activities,
controls or constraints on the activities, and mechanisms
which perform the activities. From the IDEF0 model we
generated a data dictionary consisting of the entities that
are the inputs, outputs, and controls of the activities in
the model. We used these to define the object-oriented
architecture of the AMgr.

AMgr Architecture
Major AMgr objects include System Agents, Actor
Agents, Goal Agents, Function Agents, an Agenda
Agent, and an Agenda Manager Interface. Each agent is
a simple knowledge-based object representing the
corresponding elements of the cockpit environment. As a
representative of such an element, the Agent's purpose is
to maintain timely information about it and to perform
processing that will facilitate AMgt. An Agent's
declarative knowledge is represented using instance
variables. Its procedural knowledge is represented using
Smalltalk methods. The categories of Agents are
described below and the overall architecture is illustrated
in Figure 1.

The purpose of a **System Agent** (SA) is to help the pilot
(and the AMgr itself) maintain situational awareness.
Each SA represents a system in the simulated
environment, such as the aircraft, the fuel system, or
even a pilot, and receives information from that system
via an inter-process connection called a socket. An SA's
declarative knowledge includes the past, current, and
projected future state of the corresponding system. Its
procedural knowledge includes how to project future
state and how to recognize system abnormalities. This
means that an SA maintains not only current and past
system state information, but can also be called upon by
other agents (see below) to project future state
information in order to anticipate future events. It can
also recognize system faults and instantiate Goal Agents
(see below) for goals to correct them.

Actor Agents (AAs) recognize actors' goals, implicitly
and explicitly, and make them known to the rest of the
AMgr. An AA represents an actor, such as a pilot or an
automation device. As declarative knowledge, each AA
maintains information about the current state of the
corresponding actor, including his/her/its agenda. AA
procedural knowledge covers how to obtain state
information.

A very important AA is the **Flightcrew** (or pilot) **Agent**.
The Flightcrew Agent has a serial connection to a
Verbex automatic speech recognition (ASR) system.

2

This allows the pilot to declare his/her goals explicitly by short vocal utterances. The intent is to be able to recognize pilot goals primarily by monitoring air traffic control (ATC) clearance acknowledgements. That is, when a pilot acknowledges ATC clearances, he/she typically repeats the clearance back to the controller. The Flightcrew Agent, using the Verbex system, interprets these as pilot goals for the control of the aircraft. For example, heading, altitude, airspeed, and waypoint goals are declared as the pilot verbally acknowledges ATC clearances by repeating them back to the controller (the experimenter, in our study). The Verbex system "eavesdrops" on the pilot and sends a coded form of the utterance to the Flightcrew Agent which translates it and declares a goal by creating an instance of a Goal Agent.

The purpose of **Goal Agents** (GAs) is to maintain information about all actors' goals. A GA represents an actor's goal, such as one to descend to and maintain an altitude of 9,000 ft or one to crossfeed fuel from one fuel tank to another to correct an imbalance. A GA has declarative knowledge about the state of the goal to be achieved (pending, active, or terminated) and whether or not it is achieved. A GA's procedural knowledge includes how to determine if the goal is achieved and how to determine whether or not its goal is consistent with the goals of other GAs. Each GA is associated with one Function Agent.

The purpose of a **Function Agent** (FA) is to monitor whether its goal is being pursued in a correct and timely manner. An FA represents a function, which is an activity performed to achieve a goal. Each FA has declarative knowledge about the state of its function (pending, active, or terminated, like the goal) and the status of its function (how well the function is being performed and whether or not goal achievement is likely). FA procedural knowledge includes how to assess function state and status and how to assess goal and function priority based on prevailing conditions. FAs not only assess the current status of functions, but also use the prediction capabilities of SAs to project future function status.

The single **Agenda Agent** is the executive Agent which coordinates the activities of all other Agents. Its declarative knowledge consists of the current set of GAs and FAs. Its procedural knowledge includes what to do when a new GA is introduced (e.g., check it against other GAs for compatibility), what to do when a GA changes state (e.g., move it to another part of the Agenda), and how to develop overall priority ratings for the Goal/Function Agents based on importance and urgency.

Operation
As the simulator runs it sends state data to the AMgr, whose SAs maintain a situation model of the simulated aircraft and its environment. AAs monitor real or simulated actors, detect or infer goals, and instantiate GAs. GAs look for conflicts with each other and monitor SAs to see if the goals are achieved. FAs monitor the progress -- if any -- made in achieving their associated goals. The Agenda Agent prioritizes GAs and FAs and keeps track of goal conflicts. The AgendaManager Interface presents this agenda information to the pilot.

Pilot Interface
The **AgendaManager Interface** (AMI) consists of display formats for presenting agenda information to the pilot. It is illustrated in Figure 2, which shows what the pilot would see in the possible (but hopefully, very unlikely) situation depicted in the diagram in Figure 1. Each line on the AMI is a message concerning a GA and FA pair, consisting of the name of the goal and a status comment if a problem exists or is anticipated.

In the situation underlying both figures, the Fuel System Agent has detected an out-of-balance condition between the left and right fuel tanks and has instantiated a GA for the goal to remedy it, and the pilot has correctly begun crossfeeding fuel. The corresponding FA has determined that this function is being performed satisfactorily, but will require attention later to terminate fuel crossfeeding, so the AMgr message for it is white, which denotes a satisfactory status.

The pilot has received an air traffic control clearance to reduce speed to 240 knots (kt), maintain the present heading of 070 degrees, and descend to an altitude of 9,000 ft. He/she has verbally acknowledged this clearance and the AMgr has recognized these aviate (aircraft control) goals and instantiated GAs and FAs. Speed is currently too high and is not decreasing, so the AMgr speed message is amber and its comment notes the problem. The airplane's current heading is 070 degrees, so the AMgr's message for this is gray, with no explanatory comments, so as not to distract.

Although the aircraft is correctly descending towards *9,000* ft, the pilot has inadvertently set the autoflight system to descend to *8,000* ft. This goal conflict has been detected by the two GAs and is signalled by an amber-colored message.

Two other system faults have occurred. There is a fire in the left engine and the pressure in the center hydraulic subsystem has dropped below an acceptable level, and corresponding SAs have detected them and instantiated GAs for goals to correct them. As the engine fire

condition is critical, its message is displayed in red at the very top of the display. The hydraulic system fault is intermediate in priority between the flight control goals and the fuel balance goal, it is displayed in amber between them.

AgendaManager Evaluation

Objective
The purpose of the experiment was to determine any differences in AMgt performance between the use of the AMgr and the use of a model (developed in our lab) of a conventional monitoring and alerting system called the Engine Indication and Crew Alerting System (EICAS).

Method
A total of ten airline pilots participated in the experiment, with the first two being used to refine the scenarios and identify and correct problems with software and procedures.

Prior to the experiment each subject was given a brief introduction to the study, filled out a pre-experiment questionnaire, and read and signed an informed consent document. The following forty minutes were used to train the Verbex speech recognition system to recognize the subject's voice so that altitude, speed, and heading goals could be determined from ATC clearance acknowledgements. After a short break the subject learned how to fly the flight simulator using the Mode Control Panel (MCP -- the autoflight system interface), recognize and correct experimenter-induced goal conflicts and subsystem faults, interpret EICAS and AMgr displays, and alter programmed flightpaths. After a lunch break, the subject flew two 30 minute scenarios (one with EICAS, one with the AMgr), separated by a five minute break. Upon the completion of the experiment the subject answered a post-experiment questionnaire.

The primary factor investigated in the experiment was monitoring and alerting system condition (whether AMgr or EICAS was used). The experimental design was balanced in regard to the monitoring and alerting system used and the scenario (1 or 2).

We collected data for each subject on:
1. how correctly the subject prioritized within concurrent subsystem functions;
2. the average subsystem fault correction time;
3. the average time to properly program the autoflight system;
4. the percentage of goal conflicts detected and corrected;
5. the average time to resolve goal conflicts;
6. how correctly the subject prioritized concurrent subsystem and aviate functions;
7. the average number of unsatisfactory functions at any time;
8. the percentage of time all functions were satisfactory; and
9. the subject's rating of the effectiveness of each monitoring and alerting system: -5 (great hindrance) to +5 (great help).

Results
Table I shows the results obtained for each of these variables. The first three, within subsystem correct prioritization, subsystem fault correction time, and autoflight programming time, show no significant statistical differences (p-values > 0.05) across the AMgr/EICAS conditions. This is critical for the interpretation of the results in that it supports the hypothesis of the AMgr being the only cause of significant differences. For example, within subsystem prioritization performance does not differ between the two conditions. Also, once a subsystem fault is detected, the process of correcting it is identical between the two conditions. Programming the autoflight system is identical in both conditions. However, we did observe a minor practice effect for each subject between the two scenarios, i.e., they showed significant improvement in programming the autoflight system.

A key objective of the AMgr is to support the pilot in recognizing goal conflicts and to help resolve those in a timely manner. The next two variables, goal conflicts corrected percentage and goal conflict resolution time, directly reflect this, and the results clearly indicate how successful the AMgr condition achieved it. Any time a goal conflict existed, the AMgr helped the subject identify this conflict (100%) whereas with EICAS, the subjects only identified 70% of the conflicts. Also, with the AMgr the subjects were able to resolve the conflict nearly 19 seconds faster. This may have helped them achieve an overall lower level of unsatisfactory functions (AMgr: 0.64; EICAS: 0.85) by making more time available to them.

It is crucial for the pilot to recognize that primary flight control functions (i.e., aviate functions) are usually more critical than subsystem related functions. The AMgr clearly showed its strength by helping the pilots in 72% of the cases to correctly prioritize. With EICAS the pilots only achieved 46%. Last, but not least, with the AMgr the subjects were able to achieve a significantly higher percentage where all functions were performed satisfactorily (AMgr: 65%; EICAS: 52%).

4

Independent of how well an individual can perform under a given condition, it is also important that subjectively he or she finds this condition acceptable. Based on our results, the subjects' effectiveness ratings strongly support the AMgr (4.8 vs. 2.).

Discussion

The results of our investigation clearly suggest that the concept of the AgendaManager can have a very significant impact on flight crew performance, helping them in successfully managing goals, functions, and resources. In that, the AMgr represents a software tool which shows the potential for significantly reducing the probability of undetected flight crew errors. It directly builds on the success of existing crew monitoring and alerting systems (such as EICAS) by including pilot intent logic [6]. Given the industry's objective of significantly reducing the number of commercial transport accidents, the AMgr must be seen as one of the facilitating tools in this effort.

Further Research

Based on our results, we believe that there are several research paths to be explored. For example, the AMgr should be evaluated in a more realistic scenarios in a full-mission simulator. This is necessary to be sure that the effects that we saw in this evaluation were not merely artifacts of the simplified part-task environment.

During AMgr development, we experimented with a goal communication method that integrated overt communication (via clearance acknowledgement) and covert communication (via script-based intent inferencing) [3]. Although we chose to include only overt goal communication in the current version of the AMgr, covert methods offer the potential of low pilot workload and should be further investigated.

An enhancement we are currently exploring is **Fuzzy Function Agents** (FFAs). Function Agents in the current version of the AMgr use conventional (crisp) logic to assess how well functions are being performed. In some cases (for example, aviate functions) fuzzy logic may be more appropriate, so we are developing FFAs to provide more human-like function assessments. Through interviews with pilots we extracted fuzzy if-then rules to model human function assessment. Then we fine-tuned the rules with the application of a genetic algorithm which minimized the discrepancy between human and machine assessments of sample scenarios. Although a preliminary evaluation of the FFAs has revealed performance comparable to that of human pilots, the method needs further development.

Although the AMgr has potential as an operational aid, its near-term benefits may be realized in other ways. For example, with suitable modifications, the AMgr could be embedded in a part-task trainer to facilitate AMgt training. Another possible role is as a research tool. With relatively minor changes the AMgr could be used to capture AMgt data on-line in full-mission simulator experiments. In fact, the greatest value of the AMgr may be in this capacity, helping us understand the phenomenon of Agenda Management better.

Acknowledgements

References

[1] Aeronautica Civil of the Republic of Colombia, *Controlled Flight Into Terrain, American Airlines Flight 965, Boeing 757-223, N651AA, Near Cali, Colombia, December 20, 1995*, Santafe de Bogota, DC, Colombia: Aeronautica Civil of the Republic of Colombia, September 1996.

[2] Aircraft Accident Investigation Commission - Ministry of Transport Japan, *China Airlines Airbus Industrie A300B4-622R, B1816 Nagoya Airport April 26, 1994*, Ministry of Transport, July 19, 1996.

[3] Cha, W.C. & Funk, K., "Communicating Pilot Goals to an Intelligent Cockpit Aiding System", *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, 12 - 15 October 1997, Orlando, Florida.

[4] Chou, C.D., Madhavan, D., & Funk, K.,"Studies of Cockpit Task Management Errors," *International Journal of Aviation Psychology, 6*(4), 1996, pp. 307-320

[5] Funk, K., "Cockpit Task Management: Preliminary Definitions, Normative Theory, Error Taxonomy, and Design Recommendations, *The International Journal of Aviation Psychology*, Vol. 1, No. 4, 1991, pp. 271-285

[6] Funk, K. & Braune, R., "Expanding the Functionality of Existing Airframe Systems Monitors: The AgendaManager," *Proceedings of the Ninth International Symposium on Aviation Psychology*, 27 April – 1 May 1997, Columbus, Ohio, in press.
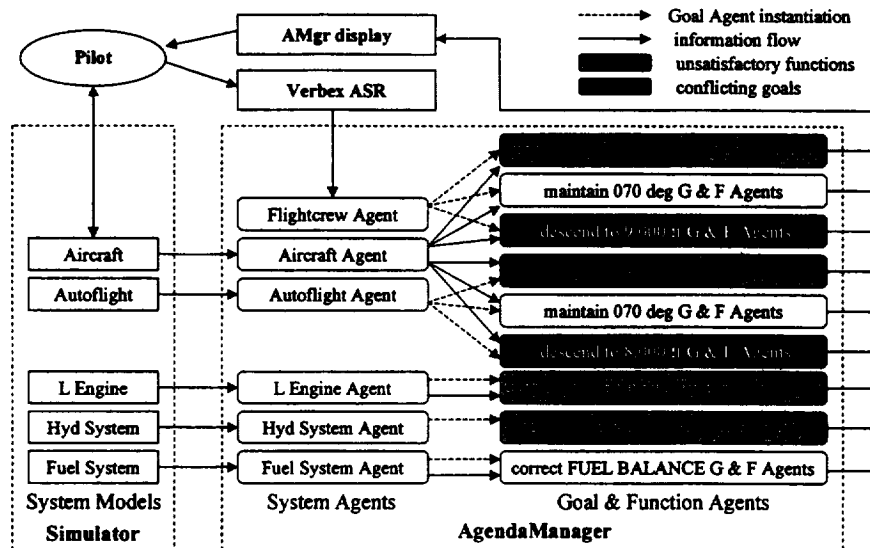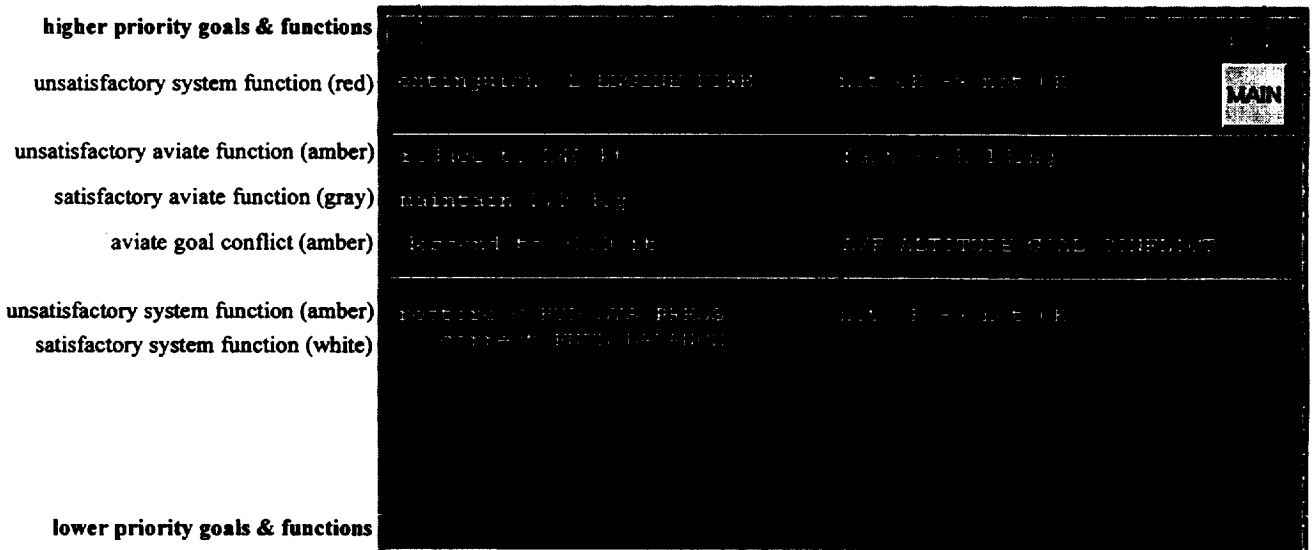
Figure 1. AgendaManager architecture.

Figure 2. AgendaManager interface.

Table I

AgendaManager evaluation results, mean values (all times in seconds).

Response variable	AgendaManager	EICAS	p-value
within subsystem correct prioritization	100%	100%	NA
subsystem fault correction time	19.5	19.6	.9809
autoflight system programming time	7.0	5.9	.1399
goal conflicts corrected percentage	100%	70%	.0572
goal conflict resolution time	34.7	53.6	.0821
subsystem/aviate correct prioritization	72%	46%	.0308
average number of unsatisfactory functions	0.64	0.85	.0466
percentage of time all functions satisfactory	65%	52%	.0254
subject effectiveness rating (-5 to 5)	4.8	2.5	.0006

Agenda Management:
Understanding and Facilitating the Management of Flightdeck Activities

Aviate
Navigate
Communicate
Manage Systems

Agenda Management

Synopsis: This website describes our efforts to understand how pilots manage -- and mismanage -- flightdeck (cockpit) activities, and how to facilitate that process through computational aids. This page provides an overview of the research with hyperlinks to more detailed descriptions.

Keywords: cockpit task management, agenda management, strategic workload management, human factors, ergonomics, aviation, flightdeck, flight deck, aviation safety, flight safety

Last update: 30 Jun 97

Ken Funk, **Woo Chang Cha,**
Robert Wilson, and Joachim Zaspel
Department of Industrial and
Manufacturing Engineering
118 Covell Hall, Oregon State
University
Corvallis, OR 97331-2407, USA
541-737-2357
funkk@engr.orst.edu

Rolf Braune
7250 Old Redmond Road,
#140
Redmond, WA 98052, USA
204-885-1943
71611.1126@compuserve.com

Joong Nam Kim
Korea Electric Power
Research Institute
Taejon, South Korea
jnkim@kepri.re.kr

Contents/Overview

Introduction

Commercial air transportation has an admirable safety record, yet each year hundreds of lives and hundreds of millions of dollars worth of property are lost in air crashes in the United States alone. About two-thirds of these aircraft accidents are caused, in part, by pilot error. Many of these errors are errors in *performing* flightdeck (or cockpit) functions, others are errors in *managing* flightdeck goals and the functions to achieve those goals. This website describes the development of a theory of flightdeck activity management and the development and evaluation of a prototype computational aids to facilitate it.

Background

Multitasking

The modern flightdeck (or cockpit) is a multitask environment. The flightcrew (whether one or more pilots) is constantly faced with multiple, concurrent, competing, often conflicting goals to accomplish and therefore must engage in multiple activities to accomplish them. As most pilots are aware, it is not only difficult to successfully *accomplish* such goals, it is often even more challenging to *manage* the activities directed towards them. We discovered this ourselves as we developed and evaluated the Task Support System (TSS), part of an experimental avionics system to aid military pilots.

Over the years, pilots have developed a priority scheme to facilitate this management of flightdeck activities:

1. **aviate** to keep the airplane in the air and pointed in the right direction;
2. **navigate** to determine where to go and how to get there;
3. **communicate** with the rest of the flightcrew and with air traffic control; and
4. **manage systems**, like engines, fuel systems, and hydraulic systems.

Cockpit Task Management

Although the process of managing flightdeck activities is intuitively well-understood by pilots, we formalized it in a preliminary, normative theory, which we called Cockpit Task Management (CTM). Briefly, a goal is a desired

behavior of the aircraft and a task is an activity performed to achieve it. As there are generally multiple, concurrent tasks to attend to on the flightdeck, the flightcrew must **create an initial list of tasks** to perform then continually

- **assess the current situation;**
- **activate new tasks** in response to recent events;
- **assess task status** to determine if each task is being performed satisfactorily;
- **terminate tasks** with achieved or unachievable goals;
- **assess task resource requirements** (human and machine);
- **prioritize active tasks;**
- **allocate resources to tasks** in order of priority (initiating, interrupting, and resuming them, as necessary); and
- **update the task list.**

To better understand the nature and significance of CTM, we conducted three empirical studies: a review of National Transportation Safety Board aircraft accident reports, a review of Aviation Safety Reporting System aircraft incident reports, and a simulator experiment. In the accident report study, we determined that CTM errors occurred in 76 (23 per cent) of the 324 accidents we reviewed. We found CTM errors in 231 (49 per cent) of the 470 incident reports we reviewed. In the simulator study we found that CTM performance was inversely related to workload. We concluded that CTM is significant to flight safety.

The Cockpit Task Management System

Although there are many potentially effective responses to this, we chose to investigate the use of computational aids to facilitate CTM. Our first such aid (not counting the TSS, which actually preceded the development of the concept of CTM) was the Cockpit Task Management System (CTMS). Our goals for the CTMS were that it should help the pilot initiate, monitor, prioritize, and terminate tasks. To achieve these goals, we determined that the CTMS should provide information about task state (upcoming, active, terminated), status (satisfactory or unsatisfactory performance), and priority.

We implemented the CTMS using Smalltalk, an object-oriented computer programming language. We used concepts of object-oriented design and distributed artificial intelligence in the CTMS implementation, where aircraft subsystems and flight tasks were represented by conceptual software units referred to as agents. In the CTMS, aircraft subsystems and pilot tasks were represented by system agents (SAs) and task agents (TAs), respectively.

Each TA used information from SAs and its own procedural knowledge to determine the state of its task: latent (not imminent), upcoming (imminent), in progress, suggested (requiring immediate attention), or finished. Task status (satisfactory or unsatisfactory) was determined in a similar way. The CTMS display provided state, status, and priority information about each task.

We performed a part-task simulator experiment to evaluate the effectiveness of the CTMS in facilitating CTM performance. Twelve subjects flew a part-task simulator under both aided (with CTMS) and unaided (without the CTMS) conditions. When subjects flew with the assistance of the CTMS, the mean task misprioritization rate was reduced by 41 per cent, the mean subject response time was reduced by 18 per cent, mean unsatisfactory aircraft control was reduced by 24 per cent, and the average number of incomplete tasks during simulator flights was reduced by 82 per cent.

Agenda Management

Our theory of CTM, as originally formulated, failed to address two important issues. First, human pilots are coming to depend more and more on automated aids, such as autopilots and centralized monitoring and alerting systems, to aid them in the monitoring and control of the aircraft and its subsystems. As machines perform certain

goal-directed flightdeck activities, it is more appropriate to speak of those activities as functions since, technically speaking, a task is a function performed by a human. Second, with both humans and machines performing flightdeck functions, there is a potential for conflicting goals. Two recent aircraft accidents illustrate such goal conflicts. In 1994 in a China Airlines Airbus A300 on approach to Nagoya, Japan, the flightcrew inadvertently initiated an autoflight system go-around maneuver while trying to continue the landing. The goal conflict between the flightcrew and the autoflight system caused an out-of-trim condition that resulted in a stall and crash which killed 264 persons. In an American Airlines Boeing 757 on approach to Cali, Columbia in 1995, the flightcrew accepted an air traffic control clearance to fly direct to a designated navigational fix. They inadvertently configured the aircraft's flight management system to fly the airplane to a different fix. This goal conflict was not detected in time to prevent the aircraft from crashing into mountainous terrain, killing 159 persons.

To address these issues, which were clearly related to the original theory of CTM, we expanded the theory. Since an 'agenda' is a list of things to do, we called the new concept **Agenda Management (AMgt)**. To formalize the concept, we developed a model of AMgt using IDEF0, a functional modeling language. IDEF0, whose name stands for ICAM (Integrated Computer Aided Manufacturing) DEFinition language 0, is a graphical modeling language. IDEF0 diagrams consist of boxes representing activities and arrows representing inputs and outputs to and from those activities, controls or constraints on the activities, and mechanisms that perform the activities. In an IDEF0 model of a process, each box represents an activity or function, which transforms its inputs to its outputs, subject to certain controls or constraints, by means of a set of mechanisms. The following summary theory of AMgt is based on the model.

An **actor** is an entity that *does* something in that it can control or change the state of the aircraft and/or its subsystems. Pilots are human actors; machine actors include autoflight and flight management systems. A **goal** is a representation (mental, electronic, or even mechanical) of an actor's intent to change the state of the aircraft or one of its subsystems in some significant way, or to maintain or keep the aircraft or one of its subsystems in some state. For example, a pilot might have a goal to descend to an altitude of 9,000 ft, a goal to maintain the current heading of 270° , and a goal to crossfeed fuel to correct a fuel system imbalance. If configured properly, the autoflight system in this example would also have a goal to descend to 9,000 ft and a goal to hold 270° . Goals come about as a result of planning and decision making in the case of human actors, and computation or human input, in the case of machine actors.

A **function** is an activity performed by an actor to achieve a goal. That activity may directly achieve the goal or it may produce sub-goals which, when achieved by performing sub-functions, satisfy the conditions of the original goal. Actors use **resources** to perform functions. Human actor resources include eyes, hands, memory, and attention; machine actor resources include input and output channels, memory, and processor cycles. Other machine resources include flight controls, electronic flight instrument system displays, and radios. In general, several goals might exist at any time, so several functions must be performed concurrently to achieve them. Actors must be assigned to perform those functions and resources must be allocated to enable them. An **agenda** then is a set of goals to be achieved and a set of functions to achieve those goals.

Agenda Management (AMgt) is a high-level flightdeck function performed cooperatively by flightdeck actors, which involves two sub-functions:

1. **Goal management** is the process of
    1. recognizing or inferring the goals of all flightdeck actors;
    2. canceling goals that have been achieved or are no longer relevant;
    3. identifying and resolving conflicts between goals; and
    4. prioritizing goals consistently with safe and effective aircraft operation.
2. **Function management** is the process of
    1. initiating functions to achieve goals;
    2. assigning actors to perform functions;
    3. assessing the status of each function (whether or not it is being performed satisfactorily and on time);
    4. prioritizing those functions based on goal priority and function status; and

5. allocating resources to be used to perform functions based on function priority.

At any point in time, **AMgt performance** is satisfactory if and only if:

1. there are no goal conflicts;
2. all goals and functions are properly prioritized; and
3. *either*
     1. performance of all functions is satisfactory, *or*
     2. if that is not possible, actors are actively engaged in bringing the highest priority unsatisfactory functions up to a satisfactory level of performance.

The AgendaManager

From the results of our CTM studies and our analysis of the Nagoya, Cali, and other aircraft accidents, we have concluded that AMgt -- and specifically the failure to perform AMgt satisfactorily -- is a significant factor in flight safety. The objectives of our most recent research task were to develop and to evaluate an experimental computational aid to facilitate AMgt. We call this aid the AgendaManager (AMgr).

Simulator

The part-task flight simulator that provides the context for the AMgr models a generic, twin engine transport aircraft. It is built from components developed at the NASA Langley and NASA Ames Research centers and in our own lab. It runs on one or two Silicon Graphics Indigo 2 computers and provides a simplified aerodynamic model (Langley), autoflight system (Langley), Flight Management System (Langley), primary flight displays (Ames), Mode Control Panel (Ames), and system models and system synoptic displays (OSU). The software is written in C, FORTRAN, and Smalltalk (VisualWorks 2.5).

Architecture and Function

From the IDEF0 model of AMgt we generated a data dictionary consisting of the entities that are the inputs, outputs, and controls of the activities in the model. We used this information to define the object-oriented architecture of the AMgr and the functions of its components. Major AMgr objects include System Agents, Actor Agents, Goal Agents, Function Agents, an Agenda Agent, and an Agenda Manager Interface. Each Agent is a simple knowledge-based object representing the corresponding elements of the cockpit environment. As a representative of such an element, the Agent's purpose is to maintain timely information about it and to perform processing that will facilitate AMgt. An Agent's declarative knowledge is represented using instance variables. Its procedural knowledge is represented using Smalltalk methods.

System Agents (SAs) represent systems modeled in the flight simulator, remembering their state and recognizing abnormal conditions such as malfunctions. System Agents provide situation information to the other AMgr Agents. **Actor Agents (AAs)** recognize actor (pilot or autoflight system) goals and instantiate Goal Agents. The Flightcrew Agent recognizes pilot goals by means of a Verbex VAT31 automatic speech recognition system as the pilot acknowledges air traffic control clearances. **Goal Agents (GAs)** represent actor goals. They detect conflicts and determine when goals are achieved. **Function Agents (FAs)** monitor the progress of activities directed towards the goals, noting whether that progress is satisfactory or unsatisfactory. The single **Agenda Agent** contains and coordinates the other Agents, introducing new Agents to its collections, checking GAs against each other to identify conflicts, and ordering Goal and Function Agents by priority. The AgendaManager Interface displays AMgt information to the pilot.

Operation

As the simulator runs it sends state data to the AMgr, whose SAs maintain a situation model of the simulated

aircraft and its environment. AAs monitor real or simulated actors, detect or infer goals, and instantiate GAs. GAs look for conflicts with each other and monitor SAs to see if the goals are achieved. FAs monitor the progress -- if any -- made in achieving their associated goals. The Agenda Agent prioritizes GAs and FAs and keeps track of goal conflicts. The AgendaManager Interface presents this agenda information to the pilot.

Evaluation

We conducted an evaluation study to compare the effectiveness of the AMgr in facilitating AMgt with that of a model of an existing aiding system called the Engine Indication and Crew Alerting System (EICAS). Eight airline pilots flew the simulator in 30-minute scenarios under two conditions, one using the AMgr, the other using EICAS. We measured several types of performance, including how well subjects detected and resolved goal conflicts and how well they prioritized goals and functions. We also asked the subjects to rate the perceived effectiveness of the two systems in aiding their performance.

For all measures where AMgr and EICAS were functionally equivalent, there was no statistically significant difference in subject performance between the condition with the AMgr and that with EICAS. For all measures where AMgr and EICAS functionality differed significantly, AMgt performance was better with the AMgr than with EICAS, and the subjects rated AMgr effectiveness higher than EICAS effectiveness. All such differences were statistically significant at the *alpha* = 0.1 level. Four were statistically significant at the *alpha* = 0.05 level.

Discussion

AgendaManager Performance

The first set of findings (that there was no difference in measures related to functionally similar capabilities) is suggestive evidence that there was no experimenter-induced bias in favor of the AMgr. The second set of findings is strong evidence that the AMgr actually facilitated AMgt in the context of this experiment.

We must, however, be cautious concerning any inferences made from this finding. The fidelity of the simulator was fairly low and the fact that we observed a period effect (which could include learning) is an indication that perhaps the subjects did not receive adequate training. The simulator was a one-pilot version whereas all of our subjects fly on a two-pilot flightdeck. Finally, the success of the AMgr depends to a very large extent on its ability to correctly recognize the pilot's goals. In five to 10 percent of our subjects' goals the automatic speech recognition system (an old model) did not recognize the goal from the subject's utterance and the Goal Agent had to be instantiated by the experimenter.

Nevertheless, our findings are suggestive that AMgt performance, which is significant to flight safety, can be enhanced by means of a computational aid. Especially in light of recent advances in automatic speech recognition technology and the Federal Aviation Administration's plans to introduce datalink technology to deliver clearances to aircraft, we believe that further development of the AMgr is warranted.

Related Systems

The relationship of the AMgr to several existing aiding systems should be noted. First, the AMgr can be considered a logical extension of the Engine Indication and Crew Alerting System (EICAS) used in present-generation Boeing aircraft, and similar centralized monitoring and alerting systems in other aircraft. EICAS and related systems have been very successful and well received by the operational community. However, they are limited in the extent to which they can tailor the information to the phase of flight and they are not capable of merging the information in case of multiple failures. Of much greater significance is that little or no effort is made to consider the flightcrew's intent at any given moment. The AMgr builds on the success of EICAS by adopting EICAS display philosophy and coding and overcomes the latter limitation by basing its operation on the pilot's declared goals.

The AMgr also has some affinity to Pilot's Associate, Rotorcraft Pilot's Associate, and CASSY (Cockpit Assistant System), all of which are aiding systems designed to offer integrative and active assistance to the pilot. The AMgr is distinguished from these and similar systems in that it does not attempt to be a general, active aid. Rather, the AMgr focuses on passively assisting the flightcrew in performing AMgt by supplementing human memory and attention, not action.

Further Research

Work remains to be done on the AMgr and the concept of AMgt. For example, the AMgr should be evaluated in a more realistic scenarios in a full-mission simulator. This is necessary to be sure that the effects that we saw in this evaluation were not merely artifacts of the simplified part-task environment.

During AMgr development, we experimented with a goal communication method that integrated overt communication (via clearance acknowledgement) and covert communication (via script-based intent inferencing). Although we chose to include only overt goal communication in the current version of the AMgr, covert methods offer the potential of low pilot workload and should be further investigated.

An enhancement we are currently exploring is **Fuzzy Function Agents (FFAs)**. Function Agents in the current version of the AMgr use conventional (crisp) logic to assess how well functions are being performed. In some cases (for example, aviate functions) fuzzy logic may be more appropriate, so we are developing FFAs to provide more human-like function assessments. Through interviews with pilots we extracted fuzzy if-then rules to model human function assessment. Then we fine-tuned the rules with the application of a genetic algorithm which minimized the discrepancy between human and machine assessments of sample scenarios. Although a preliminary evaluation of the FFAs has revealed performance comparable to that of human pilots, the method needs further development.

Although the AMgr has potential as an operational aid, its near-term benefits may be realized in other ways. For example, with suitable modifications, the AMgr could be embedded in a part-task trainer to facilitate AMgt training. Another possible role is as a research tool. With relatively minor changes the AMgr could be used to capture AMgt data on-line in full-mission simulator experiments. In fact, the greatest value of the AMgr may be in this capacity, helping us understand the phenomenon of Agenda Management better.

Acknowledgements