1999-01-5536

The AgendaManager: A Knowledge-Based System to Facilitate the Management of Flight Deck Activities

Ken Funk
Oregon State University

Rolf Braune
Braune and Associates, Inc.

1999 World Aviation Conference
October 19-21, 1999
San Francisco, CA

SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL

SAE International
400 Commonwealth Drive
Warrendale, PA 15096-0001 U.S.A.

AIAA
American Institute of Aeronautics and Astronautics
370 L'Enfant Promenade, S.W.
Washington, D.C. 20024

Published by the American Institute of Aeronautics and Astronautics (AIAA) at 1801 Alexander Bell Drive, Suite 500, Reston, VA 22091 U.S.A., and the Society of Automotive Engineers (SAE) at 400 Commonwealth Drive, Warrendale, PA 15096 U.S.A.

Produced in the U.S.A. Non-U.S. purchasers are responsible for payment of any taxes required by their governments.

ISSN #0148-7191

All AIAA papers are abstracted and indexed in International Aerospace Abstracts and Aerospace Database.

All SAE papers, standards and selected books are abstracted and indexed in the Global Mobility Database.

Copies of this paper may be purchased from:

AIAA's document delivery service
Aeroplus Dispatch
1722 Gilbreth Road
Burlingame, California 94010-1305
Phone: (800) 662-2376 or (415) 259-6011
Fax: (415) 259-6047

or from:

SAExpress Global Document Service
c/o SAE Customer Sales and Satisfaction
400 Commonwealth Drive
Warrendale, PA 15096
Phone: (724) 776-4970
Fax: (724) 776-0790

SAE routinely stocks printed papers for a period of three years following date of publication. Quantity reprint rates are available.

1999-01-5536

The AgendaManager: A Knowledge-Based System to Facilitate the Management of Flight Deck Activities

Ken Funk
Oregon State University

Rolf Braune
Braune and Associates, Inc.

ABSTRACT

Cockpit Task Management (CTM) is the process by which pilots selectively attend to tasks in such a way as to achieve their mission goal. Through our research we have found that CTM is a significant factor in flight safety, at least partly accounting for a substantial number of aircraft incidents and accidents. We developed an experimental knowledge-based system called the AgendaManager to facilitate Agenda Management (a superset of CTM) and demonstrated its superiority to a conventional crew monitoring and alerting system in a controlled evaluation study. The success of the AgendaManager is attributable not to its use of artificial intelligence technology. Rather, it is effective because it was developed using a sound human factors research and development approach. This approach and its application in AgendaManager development are the topics of this paper.

INTRODUCTION

Knowledge-based systems have a potentially significant role to play in improving the safety and effectiveness of commercial transport aircraft. However, we must overcome the temptation to let artificial intelligence technologies themselves determine the goals and guide the course of the research and development process. Rather, knowledge-based systems will find their most effective implementation if their development is needs-driven rather than technology-driven.

This paper presents an argument for a human factors approach to improving commercial transport safety and effectiveness and describes an example of the application of that approach in the development of a knowledge-based system called the AgendaManager.

IMPROVING AVIATION SAFETY THROUGH HUMAN FACTORS RESEARCH AND DEVELOPMENT

It is common knowledge that most (60% - 70%) commercial transport aircraft accidents are due in large part to flightcrew error (e.g., Boeing, 1998). This means that the greatest opportunity to reduce accidents and improve safety will come from addressing the factors -- human factors -- that contribute to these errors. Human factors engineering (also called ergonomics) is concerned with improving system safety and effectiveness by giving explicit consideration to human operator characteristics, capabilities, and limitations.

Human factors research and development seeks first to understand those human factors that contribute to accidents, incidents, and other undesirable events, then to select or develop and implement appropriate technology (hardware, software, procedures, etc.) to address those factors and improve safety and effectiveness.

Figure 1 is a highly simplified model of an idealized human factors research and development process designed to improve the safety and effectiveness of the air transportation system. Due to space limitations, certain details are omitted, but the following sections describe the major elements of the process. For each process element, we summarize its inputs, the process itself, and its outputs. Following this general overview of human factors research and development, we provide an example of its application in the creation of a knowledge-based system called the AgendaManager.

OBSERVATION – The impetus for technological development should come from the need to improve something. In this case we are interested in improving the safety and effectiveness of the commercial air transportation system. So the first step is careful, informed, and systematic observation of that system.

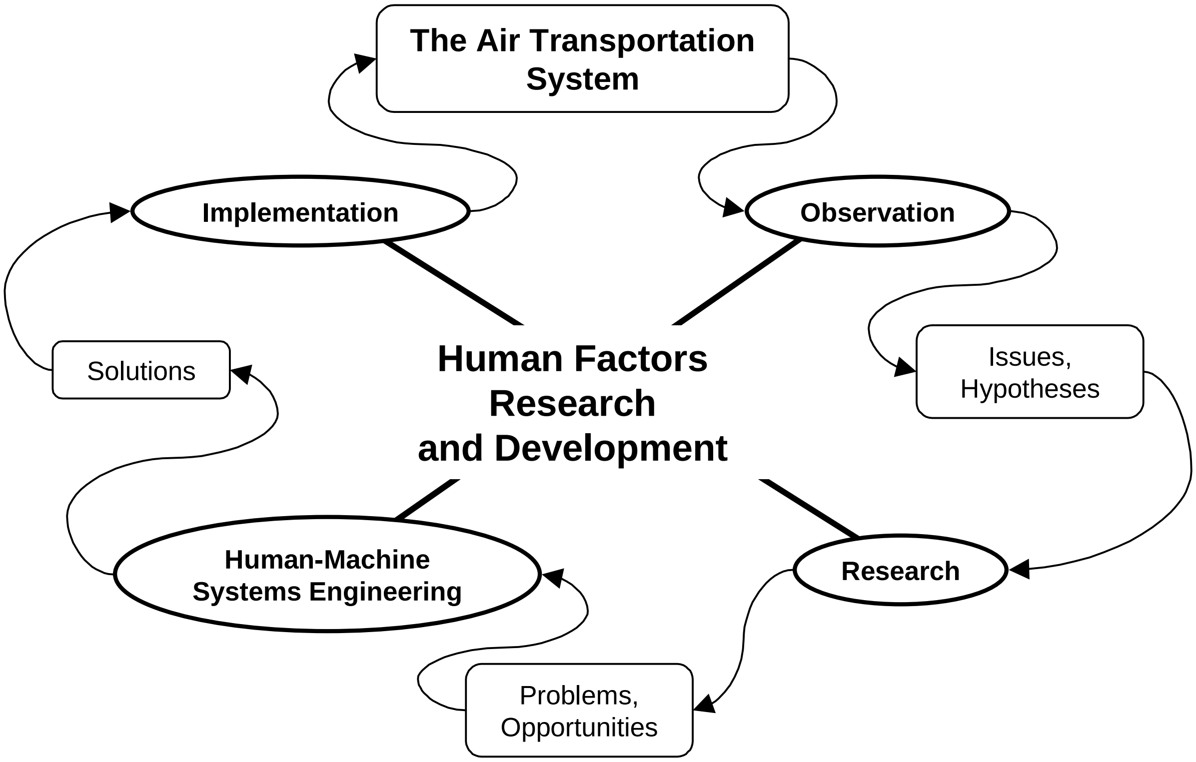

Figure 1. The Human Factors Research and Development Process.

Observation Input – In particular, we are interested in observing, recording, and interpreting significant events that occur in the system. **Accidents** are the most significant and tragic events that occur and, fortunately, they are rare. However, if Boeing's and Airbus's worldwide traffic growth projections of 5%, compounded annually, hold up, one major hull loss accident per week will occur by 2014 (Flight Safety Foundation, 1998). This projection should give us even greater incentive to observe and learn.

Besides accidents, we should note and record **incidents** -- near-accidents -- that give more representative insights into flight operations and the factors that lead to flightcrew error. We also can and must learn from other common events: **operational difficulties**, including delays, inefficiencies, uncomfortable rides, and flightcrew errors.

The Observation Process – Since the commercial air transportation system is so large and complex, the process of observing it to recognize, record, and interpret significant events tends to be informal and intuitive, and by no means is or should it be practiced solely by human factors scientists and engineers. But human factors engineering can and should contribute to the process by bringing to it knowledge about human operator characteristics, capabilities, and limitations. Given the contribution that human error makes to aircraft accidents, this knowledge is essential in the *interpretation* of the events that are observed.

Observation Output – This informed interpretation yields two important outputs. First, observation should lead to the formulation of clearly stated **issues**: concerns or possible problems with equipment or procedure design and with human performance, and therefore system safety and effectiveness.

Second, to the extent that these issues and concerns are potentially well-founded, observation should lead to **hypotheses** about what is wrong, why it is wrong, and what can be done about it.

RESEARCH – Human factors research seeks to investigate these issues to determine if they are really problems that require solutions and, if so, to better understand the problems and to create or identify opportunities to solve those problems.

Research Input – Besides the issues and hypotheses produced by the observation process, inputs to the research process include **accident and incident reports**, **operational data**, and **findings from previous research**.

The Research Process – The research process can take many forms. An important tool is accident report analysis. In the United States, the US National Transportation Safety Board (NTSB -- see their website at http://www.ntsb.gov/) conducts rigorous investigations of all commercial transport aircraft accidents and publishes accident investigation reports. Thorough and systematic studies of NTSB findings from sets of carefully selected accident reports can discover evidence to support or refute issues and hypotheses.

Due to the large numbers of aircraft incident reports available from the US Aviation Safety Reporting System (ASRS -- see their website at http://www-afo.arc.nasa.gov/ASRS/ASRS.html) and similar systems elsewhere, there is a wealth of operational information regarding human factors and flightcrew error. Since incident reports are generally voluntary, anonymous, and relatively unstructured, incident analysis is fraught with methodological difficulties. Nevertheless, rigorous, systematic analyses based on well-thought-out procedures and appropriate statistical methods can yield evidence related to issues and hypotheses.

Another research method useful for investigating issues and hypotheses is direct observation. Well designed jumpseat observations can generate useful data.

An approach growing in popularity is the Flight Operational Quality Assurance Program (FOQA -- Flight Safety Foundation, 1998). Properly analyzed (and that is no trivial challenge), FOQA data has great potential to help give human factors scientists and engineers a better understanding of issues relating to human performance and evidence related to hypotheses about the human factors that contribute to that performance.

As valuable as accident, incident, and operational data are, the overwhelming complexity of the commercial transport flight deck (cockpit) environment makes controlled investigation -- the standard means of scientific research -- extremely difficult. To partly overcome that difficulty, human factors scientists can and should turn to simulator experiments to explore issues and test hypotheses. A full-mission simulator offers high levels of fidelity along with the opportunity to control selected variables necessary to controlled experimentation. Where even simulators are too complex to tease out information about fundamental human capabilities and limitations, part-task simulator and other laboratory experiments can and should be used.

Research Output – The outputs of the research process are many. First, by generating and compiling evidence related to issues suggested by observation, research can confirm, with reasonable certainty, that there are certain human factors **problems** that require solutions or human performance **challenges** that must be met.

Research can also yield **opportunities** for solving problems and meeting challenges. In some cases, these opportunities arise from the creation of new technologies or the identification and novel application of existing technologies: hardware and/or software to augment human capabilities and overcome human limitations. In other cases, these opportunities take the form of new methods, techniques, policies, and procedures to enhance human performance.

HUMAN-MACHINE SYSTEMS ENGINEERING – Identifying and confirming problems that compromise safety and effectiveness and identifying potential solution opportunities through good research are important steps, but engineering is necessary to arrive at effective solutions that can be implemented. Human-Machine Systems Engineering (HMSE) is a systematic, disciplined approach to developing such solutions. Rather than being driven by new technologies that someone merely feels might be interesting to apply in a new domain, HMSE starts out with a thorough analysis of operational needs and requirements and systematically designs effective solutions using appropriate technologies. HMSE is therefore a **needs-driven approach** rather than a technology-driven approach.

Though there is no "standard" procedure for this approach (and, in fact the term 'Human-Machine Systems Engineering' is by no means universally used), the summary we present below is based on widely accepted sources (e.g., Anonymous, 1979, 1987; Booher,1990; Chapanis, 1996; SAE, 1995).

HMSE Input – Besides the **problems and opportunities** identified by the research process, there are two other important inputs to the HMSE process. First, the actual **research findings** themselves often offer important insight into needs, requirements, and appropriate technologies from which to develop solutions. Second, **operational knowledge** is an essential input to the HMSE process. The collective experience and knowledge of representative system participants (chief pilots, instructor pilots, line pilots, even air traffic controllers) are necessary ingredients and must be solicited and considered early and often in HMSE process.

The HMSE Process – The HMSE process starts with a document called a **statement of need** (SON). While the SON often comes from *perceived* customer desires, it should be based at least as much on science as on opinion and therefore should be built around the problems and opportunities identified in the research process.

Next comes **requirements definition**. This involves the careful and systematic identification and documentation of the environment in which the human-machine system will operate, the characteristics of its users and operators, and the performance requirements of the system. Operational knowledge as well as input from potential customers and regulatory agencies are especially important to this part of the process.

The workstation (e.g., flight deck or cockpit) and human-machine interface (e.g., controls and displays) exist in the context of the larger system (e.g., the aircraft) and before the human factors engineer can go farther, that context must be understood. **System analysis** is the process of describing and documenting -- in detail -- those parts of the larger system (e.g., engines, hydraulics, etc.) which the human must be able to monitor and control.

Knowing what the human must be able to interact with is part of the contextual picture. In addition, that system must be operated in the context of a mission. **Mission analysis** is the process of describing and documenting typical -- and atypical -- missions. In it the human factors engineer tries to identify representative scenarios that the human-machine system may encounter and by doing so determine the human performance requirements necessary to achieve adequate levels of system safety and effectiveness.

A function is a process performed to achieve a goal. **Function analysis** involves a detailed decomposition of human-machine system functions. Starting with the top-level, mission function (e.g., to transport passengers and cargo to an intended destination safely and on time), function analysis breaks the mission down into finer and finer elements to determine what functions must be performed to satisfactorily complete the mission under all foreseeable scenarios.

Function allocation is the assignment of some low-level functions to humans and other low-level functions to machines. Here the human factors engineer must balance human and machine capabilities and limitations against technical feasibility and economic constraints. It is important, however, not to simply automate every function that can be automated, leaving the remainder to the human, but to allocate functions in a manner consistent with human characteristics in the context of the system and mission as determined and documented in the previous steps.

A task is a function assigned to a human and **task analysis** is a detailed examination of the tasks each human operator is to perform. Task analysis yields a list of information needed to perform each task, the decisions the human must make, and the control actions the human must execute.

Ideally, no commitment is made to specific human-machine interaction technologies (hardware or software) in the previous steps. It is in **basic design** that the human factors engineer begins to consider the best means for performing functions and supporting tasks. *Only when the system context is thoroughly understood (from system analysis), the tasks the human must perform are known (from function analysis and allocation) and the information, decision, and action needs have been identified (from task analysis) can the designer begin selecting appropriate technologies to integrate into a complete human-machine system.* Basic design establishes general specifications for the major elements and general layout of the workstation and human-machine interface. It is guided by the results of the previous steps in the HMSE process, by human factors principles and design guidelines, and by operational knowledge.

Detailed design involves a careful refinement of the general specifications developed in basic design in light of requirements, human factors principles, and operational knowledge, and establishes detailed specifications for the human-machine system and all its elements.

Though both design phases involve continuous examination of evolving design specifications, thorough evaluation cannot (yet, at least) be made based on design specifications only. In **prototype development**, engineers create a full-scale instantiation of the design, which may range from a static mockup to a sophisticated, part-functional prototype, perhaps embedded in a simulator.

In the **test and evaluation** phase, the prototype is evaluated to assess human-machine system performance with respect to the performance requirements established earlier in the HMSE process. The prototype is exercised in normal and non-normal scenarios developed in the mission analysis step and performance measures assessing speed, accuracy, operator satisfaction, and training time requirements are applied. If requirements are met, the design may be passed along for implementation, but often, early test and evaluation indicate that refinements must be made to the design, necessitating the revisiting of earlier steps in the process. HMSE is typically a process of iterative refinement until requirements are met.

<u>HMSE Output</u> – The outputs of the HMSE process are potential **solutions** to the problems identified earlier. These solutions may consist of several components. First, design specifications for equipment and procedures provide detailed guidance for the implementation of the solutions. Second, the prototypes developed in the process provide a starting point from which operational equipment can be developed. Third, the test and evaluation results indicate the potential effectiveness of the proposed solutions. Finally, recommendations derived from the test and evaluation phase provide guidance for implementation and/or further refinement.

IMPLEMENTATION

<u>Implementation Input</u> – The inputs to the implementation process are the **solutions** developed by the HMSE process: the design specifications and recommendations for implementation.

<u>The Implementation Process</u> – This process involves the production of equipment by manufacturers, the implementation of procedures by user organizations (e.g., airlines), and the establishment of regulations or dissemination of guidance by government regulatory agencies. However, human factors engineers can and should play a significant role in final design of equipment, policies, procedures, regulations, and guidance materials.

<u>Implementation Output</u> – When performed correctly and, assuming that the other processes were performed well, the output of the implementation process is **a safer, more effective air transportation system**.

AN EXAMPLE: THE AGENDAMANAGER

INTRODUCTION AND OVERVIEW – In the remainder of this paper, we will illustrate the application of the human factors research and development process through an example of a knowledge-based system to aid human performance called the AgendaManager. For a more complete description of our work, see our website at http://flightdeck.ie.orst.edu/CTM/

OBSERVATION: COCKPIT TASK MANAGEMENT ERRORS

Observations – Aircraft accidents are usually due, at least in part, to flightcrew error. Often, these are errors in *performing* tasks -- that is, the flightcrew performs an appropriate task, but the level of performance (speed, accuracy) is insufficient and an accident results. However, we discovered that a number of aircraft accidents could be attributed to errors pilots make in *managing* tasks: failing to start tasks at the proper time, failing to terminate tasks at the appropriate time, or attending to one task when they should have been attending to another.

For example, in the notorious L-1011 Everglades crash in 1972 which killed 99 (NTSB, 1973), the flightcrew became distracted from the primary task of controlling the aircraft's altitude by a landing gear position indicator malfunction. All three flightcrew members plus a jumpseat occupant became so absorbed in the task of diagnosing the malfunction that they failed to notice a gradual descent until impact occurred.

Of course, managing tasks is a concept intuitively well understood by pilots. From their early training, they are taught an ordering of tasks to follow in deciding to which to attend at any given time:

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

Ideally, attention is given to a task lower in the hierarchy only when higher tasks are being performed satisfactorily. Unfortunately, this management process is not always practiced satisfactorily, as the Everglades accident and many other accidents and incidents show.

Hypotheses – Based on these observations, we formulated the obvious hypothesis that to safely and effectively complete a mission, **the flightcrew must properly manage as well as correctly perform tasks**.

We elaborated this hypothesis in the form of a preliminary, normative theory (Funk, 1991), briefly summarized as follows. The cockpit is an environment, in which potentially many important tasks can simultaneously compete for pilot attention. Cockpit Task Management (CTM) is the process by which pilots selectively attend to tasks in such a way as to achieve the mission goal. It determines which of perhaps many concurrent tasks the flightcrew attends to at any particular point in time. More specifically, CTM entails initiation of new tasks, monitoring of on-going tasks to determine their status; prioritization of tasks based on their importance, status, urgency, and other factors; allocation of human and machine resources to high priority tasks; interruption and subsequent resumption of lower priority tasks; and termination of tasks that are completed or no longer relevant.

With this preliminary theory as a starting point, we posed two additional hypotheses. First, following from the observation of a number of accidents involving task management errors, we hypothesized that **CTM is a significant factor in flight safety**. Second, knowing that an understanding of a problem often yields solutions to it, we further hypothesized that **CTM can be improved**, reducing the risk of incidents and accidents resulting from task management errors.

COCKPIT TASK MANAGEMENT RESEARCH – Our next step was to set about understanding the nature and significance of CTM through a program of research.

Studies

CTM Error Taxonomy – As a prelude to our research, we developed a CTM error taxonomy consisting of the following CTM error categories. A task may be initiated too early, too late, under incorrect conditions, for incorrect reasons, or not at all. Once initiated, a task may be assigned a priority that is too high, such that it prevents or degrades the performance of a task that should be performed at that time, or too low, such that its satisfactory and timely completion is prevented or hindered. A task may be terminated too early, too late, under incorrect conditions, for incorrect reasons, or not at all. This taxonomy or a derivative thereof was used in the following studies.

Accident Report Study – Our first study was a more detailed examination of accident reports to ascertain the role of CTM errors in commercial transport aircraft accidents (Chou et al, 1996). We reviewed abstracts of 324 US National Transportation Safety Board (NTSB) aircraft accident reports from the period 1960 through 1989. We removed those obviously unrelated to the CTM study, for example, those due primarily to weather and mechanical failures. This elimination process left 76 accident reports for further analysis. Using the above CTM error taxonomy 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 per cent of the accidents reviewed.

Incident Report Studies – Besides the accident report study, we conducted several incident report studies. In each case, 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 first incident report study focussed 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 per cent and 54 per cent respectively of these reports. The high incidence of CTM errors in the CFTT reports as well as the fact that over 49 per cent of all airline accidents occur during approach and landing (Boeing, 1998), caused us to focus further attention on the terminal phases of flight. For our second incident report study we obtained 243 additional reports pertaining to these phases. From the ASRS incident reports thus obtained, we eliminated duplicates. We then reviewed the remaining 470 unique reports and found CTM errors in 231 (49%) of them.

Part-Task Simulator Study – Aircraft accidents are rare events, thus providing few opportunities for developing insights into error processes. Also, though incident reports can provide first-hand information on routine 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 above and to provide an opportunity for objective observations. That was the motivation for our first part-task simulator study of CTM.

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. In the experiments eight subjects flew a low fidelity, part-task flight simulator in scenarios in which we controlled workload, maximum number of concurrent tasks, and flight path complexity. We measured the subjects' average response time to system faults, root-mean-square (RMS) flight path error, task prioritization score (based on number of correct task prioritizations as determined using the CTM error taxonomy), and the number of tasks that were initiated late.

We obtained a number of interesting results. First, workload had a significant effect on late task initiation. Second, both workload and the combination of flightpath complexity with number of concurrent tasks created significant effects on task prioritization. In other words, we found that task prioritization performance degrades as either one of these factors increase. Finally, heading deviations were significantly affected by the combination of flightpath complexity and the number of tasks and changes in workload were significant to the altitude deviations.

Automation and CTM Study – Although not completed in time to claim it as part of the basis for the AgendaManager, preliminary results from another incident report study (Wilson and Funk, 1998) influenced its development. In this study we compared two samples of ASRS incident reports to determine if level of automation on the commercial aircraft flight deck affected the frequency of task prioritization errors. The first sample was composed of 210 incident reports submitted by pilots flying advanced technology aircraft and the second sample was composed of 210 incident reports submitted by pilots flying traditional technology aircraft. In total, we analyzed 420 incident reports. Using a methodology based on the CTM error taxonomy, we classified 43 of the 420 reports (10.2%) as containing task prioritization errors (i.e., a subset of CTM errors). Of these, 28 were from the advanced technology sample and 15 were from the traditional technology sample. We concluded that task management may be more challenging in advanced technology aircraft than in conventional aircraft.

Other Research – We supplemented our own research by reviewing the related research of others, including Rogers (1996), Damos (1997), Latorella (1996), and Schutte and Trujillo (1996). Their findings were consistent with ours: that distractions and interruptions can interfere with CTM with potentially dangerous consequences.

Summary of Findings – From reflections on our own research findings as well as those of others, we came to two conclusions. First, there is a problem: **CTM is a significant factor in flight safety**. That is, CTM errors are evident in a significant percentage of incidents and accidents. But second, there is an opportunity: **CTM can potentially be improved**. Although we did not have enough hard evidence at the time to justify a specific approach to CTM aiding, we hypothesized that significant improvements in CTM performance could be achieved by providing computer-based assistance to enhance human situation assessment and working memory.

USING HUMAN-MACHINE SYSTEMS ENGINEERING TO CREATE THE AGENDAMANAGER

Statement of Need and Requirements Definition – We formulated a Statement of Need based on our research findings (Chou et al, 1996). In summary, we set out to create a computational aid to facilitate CTM by

1. maintaining a current model of aircraft state and current cockpit tasks,
2. monitoring task state and status,
3. computing task priority,
4. reminding the flightcrew of all tasks that should be in progress, and
5. suggesting that the flightcrew attend to tasks that did not show satisfactory progress.

Though not formally stated, it was always our intent to create an aiding system that left the pilot in control by acting as a passive situation assessor and memory aid rather than as an active assistant. This was motivated by our wish to not increase workload by requiring complex interaction between the flightcrew and the aiding system.

System Analysis – Our aiding system was developed to operate in the context of a part-task simulator and although we were free to define that context, we chose to model a generic, twin-engine transport airplane. This decision was based on the large numbers of such aircraft in use today, the availability of documentation on twinjet systems, and the availability of simulator components.

Our system analysis of this simulated airplane yielded the following major subsystems: powerplant, fuel system, electrical system, hydraulic system, adverse weather system, autoflight system, and flight management system. For each subsystem we defined the major state variables that the flightcrew of such an aircraft would have to monitor and control. Those later became specifications for the simulation model components we created and assembled.

Mission Analysis – To perform mission analysis, we first decomposed a typical mission into phases then decomposed each phase into simpler activities. To assure completeness and consistency of this process as well as to yield a formal result, suitable for use in later parts of the development process, we used an analysis method called IDEF0. See Figure 2.

IDEF0 is a modeling language for representing activities in complex systems. An IDEF0 diagram (also called a node) consist of boxes and arrows. The boxes represent activities and are labeled with verb phrases. The arrows represent things (matter, energy, information, properties, etc.) that affect or are affected by those activities and are labeled with noun phrases.

The relationship of an arrow to a box represents how the thing is related to the activity. Arrows coming into the left side of the box are called inputs to the activity and represent things that are transformed by the activity. Arrows coming out from the right side of the box are called outputs from the activity and represent the results of transforming the inputs. Arrows coming into the top are called controls and represent things that constrain, limit, or guide the activity. Arrows coming into the bottom are called mechanisms and represent the things that actually perform the activity or are directly used in performing the activity.

Generically, an IDEF0 diagram is read thus: "The activity transforms its inputs to its outputs, subject to its controls and is performed by means of its mechanisms." Following this paradigm, the A4 node in the IDEF0 diagram in Figure 2 may be read thus: "*perform cruise activities* transforms the aircraft at cruise altitude to the aircraft at the top of descent point. It is constrained by air traf-

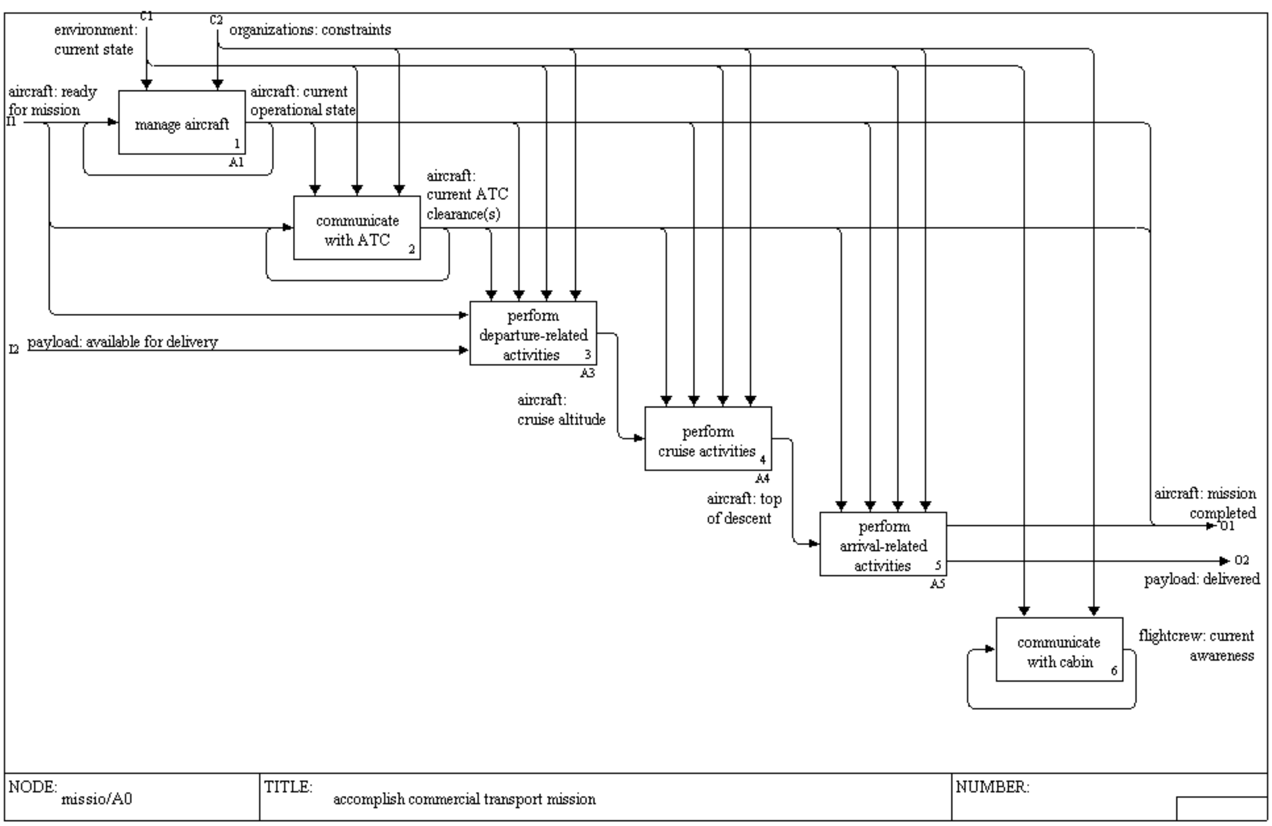

Figure 2. First level of mission analysis for AgendaManager development.

fic control (ATC) clearances, the current operational state of the aircraft, the current state of the aircraft's environment, and organizational constraints such as airline policies and procedures. Mechanisms or means for performing the activity, such as the aircraft and the flightcrew, are omitted from this diagram to reduce its complexity.

Associated with each IDEF0 diagram is a glossary which defines all labels used in the diagram. If an activity may be decomposed into simpler subactivities, its IDEF0 box may be decomposed into sub-boxes. These sub-boxes appear in a subsequent IDEF0 node.

Besides defining a typical, normal mission using IDEF0, we defined certain non-normal events that might be encountered in a mission, including equipment malfunctions and fuel imbalance conditions.

Function Analysis – While the mission analysis ultimately led to a set of detailed activities, functions, or tasks performed on a typical commercial transport mission, it did not explicitly represent the process of CTM. To gain a task management perspective on cockpit activities, we performed a separate function analysis of this management process.

Agenda Management – We found it necessary to make our function analysis address a more comprehensive process than CTM as described above. 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 cockpit 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 cockpit functions, there is a potential for conflicting goals.

To address these issues, we expanded our theory of CTM to include management of goals and management of functions -- both functions performed by humans (i.e., tasks) and functions performed by machines (e.g., autopilots). We called this expanded process Agenda Management (AMgt) to emphasize the need to manage an agenda of goals and functions. Following is a brief summary of AMgt.

An **actor** is an entity that, through its actions, controls or changes 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 degrees, 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 degrees. 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 process 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 involves the following **processes related to goals**:

- recognizing or inferring the goals of all cockpit 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.

It also involves the following **processes related to functions**:

- 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.

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.

So AMgt is a superset of CTM and we expanded our statement of need to address that superset. Rather than merely aiding the management of tasks (i.e., functions performed by humans), our system would be developed to facilitate the management of all goals and machine functions as well.

Function Analysis of Agenda Management – Towards meeting these expanded needs, we used IDEF0 to perform a function analysis of AMgt to define the functions performed in the AMgt process. Since AMgt consists of *functions to manage functions*, to avoid confusion in the remainder of this paper, we will generally refer to the functions performed in AMgt as *AMgt functions* and the aviate, navigate, communicate, and manage systems (ANCS) functions (including the ANCS tasks performed by humans) as *ANCS functions*. Figure 3 is an IDEF0 diagram of part of the results of the function analysis, showing the major AMgt functions of managing goals, managing ANCS functions, assigning actors to ANCS functions, and allocating resources to ANCS functions.

AMgt Function Allocation – Though there are formal methods for function allocation, our process of allocating AMgt functions was an informal one, guided by three criteria. First, we adopted a human-centered approach in which our aid would perform passive aiding functions rather than active controlling functions. Second, we allocated AMgt functions so as to leave the human in control of an integrated, coherent set of activities. Third, what we could automate was limited by technical feasibility.

For example, these criteria led us to allocate to our aiding system portions of the *manage goals* and *manage [ANCS] functions* AMgt functions (the arrows pointing up into the A111 and A112 AMgt functions in Figure 3). Assigning actors and allocating resources to ANCS functions were left to the human.

Task Analysis – We used the results of our mission and function analyses in our task analysis to determine what information the pilot would need to help him/her perform AMgt functions. These requirements were in turn used to design the aiding system display (see below).

Basic and Detailed Design

Object-Oriented Design – We called our aiding system the AgendaManager (AMgr). The things (e.g., systems and system states) and activities (i.e., ANCS and AMgt functions) defined in our system, mission, and function analyses made object-oriented design supplemented by human factors principles and guidelines a natural choice for designing the AMgr.

From the IDEF0 functional analysis of AMgt (see above) we generated a data dictionary consisting of the entities that are the inputs, outputs, and controls of the AMgt functions. We used this information to define the object-oriented architecture of the AMgr and the functions of its components. IDEF0 noun phrases (e.g., goal, function) became AMgr objects. IDEF0 verb phrases (i.e., the names of the AMgt functions) became methods.

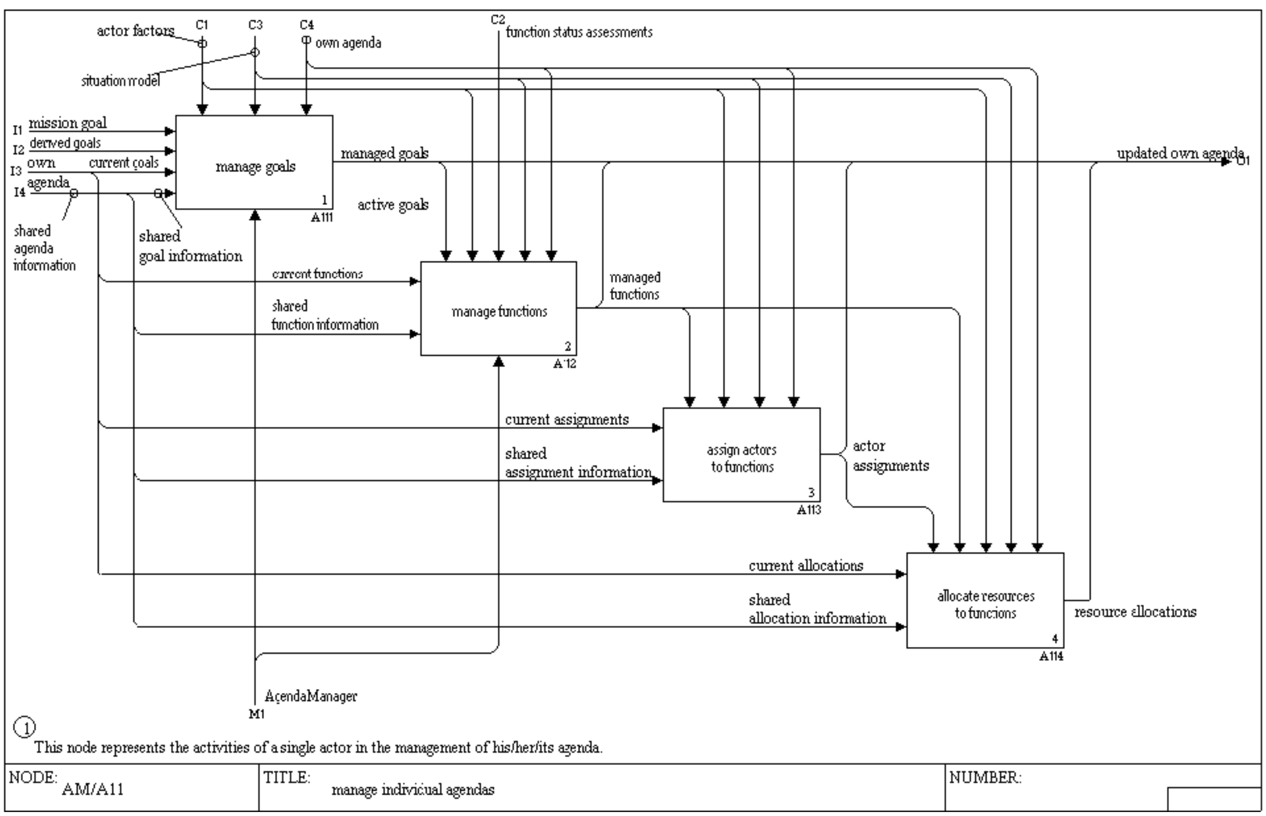

Figure 3. Part of the AMgt function analysis.

Selection of a Multiagent Approach – We designed the AMgr as a multiagent (i.e., distributed artificial intelligence or DAI) system (Weiss, 1999). It is extremely important to understand that we chose this approach not to simply demonstrate an application for multiagent/DAI technology, but because it was the right technology to use, given the findings from the previous steps in the development process. AMgt functions are complex, cognitive functions, often involving qualitative reasoning. As artificial intelligence (AI) methods were created to perform just this kind of computation, some form of AI was appropriate. AMgt as a whole involves a complex interplay of many entities (systems, actors, goals, and functions), making its facilitation through a centralized, monolithic knowledge-based system extremely difficult. Clearly, a distributed, multiagent system offered us what we believed to be the best approach.

AMgr agents included System Agents, Actor Agents, Goal Agents, Function Agents, an Agenda Agent, and an Agenda Manager Interface (i.e., display). Each Agent was designed to be a simple knowledge-based object representing the corresponding elements of the cockpit environment. As a representative of such an element, the Agent's purpose was to maintain timely information about it and to perform processing to facilitate AMgt.

AgendaManager Display Design – The AMgr display design was based on the information requirements identified in the task analysis (see above). Following a comprehensive review of display design principles, we developed two alternative display formats and compared them in a part-task simulator study (Wilson, 1997). Using the findings from the study, the display design principles, and previously established recommendations for the design of conventional monitoring and alerting systems (e.g., Boucek, Veitengruber, and Smith, 1977; Berson et al., 1981), we designed the final AMgr display.

<u>Prototype Development</u> – The prototype AMgr was developed in the Smalltalk programming language (ParcPlace VisualWorks version 2.5) and interfaced to a part-task simulator.

Simulator – The part-task flight simulator that provided the context for the AMgr modeled a generic, twin engine transport aircraft (as described above). It was built from components developed at the NASA Langley and NASA Ames Research centers and in our own lab at Oregon State University. It ran on two Silicon Graphics Indigo 2 computers and provided a simplified aerodynamic model (Langley), autoflight system (Langley), Flight Management System (Langley), primary flight displays (Ames), Mode Control Panel (Langley), and system models and system synoptic displays (OSU). The software was written in C, FORTRAN, and Smalltalk.

Architecture and Function – The multi-agent architecture of the AMgr is shown in Figure 4. Each AMgr Agent's declarative knowledge was represented using Smalltalk instance variables. Its procedural knowledge was represented using Smalltalk methods. System Agents (SAs) represented systems modeled in the flight simulator, remembering their state and recognizing abnormal conditions, such as malfunctions. System Agents provided situation information to the other AMgr Agents. Actor Agents (AAs) recognized actor (pilot or autoflight system) goals and instantiated Goal Agents. The Flightcrew Agent recognized pilot goals by means of a Verbex VAT31 automatic speech recognition (ASR) system as the pilot acknowledged air traffic control clearances. Goal Agents (GAs) represented actor goals. They detected conflicts and determined when goals were achieved. Function Agents (FAs) monitored the progress of activities directed towards the goals, noting whether that progress was satisfactory or unsatisfactory. The single Agenda Agent contained and coordinated 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 displayed AMgt information to the pilot.

Display – The AMgr display used in the prototype is shown in Figure 5.

Operation – As the simulator ran it sent state data to the AMgr, whose SAs maintained a situation model of the simulated aircraft and its environment. AAs monitored real or simulated actors, detected or inferred goals, and instantiated GAs. GAs looked for conflicts with each other and monitored SAs to see if the goals were achieved. FAs monitored the progress -- if any -- made in achieving their associated goals. The Agenda Agent prioritized GAs and FAs and kept track of goal conflicts. The AgendaManager Interface presented this agenda information to the pilot.

In the situation underlying Figures 4 and 5, 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 (Figure 4). 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 on the AMgr display (Figure 5) 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 Flightcrew Agent has recognized these aviate 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.

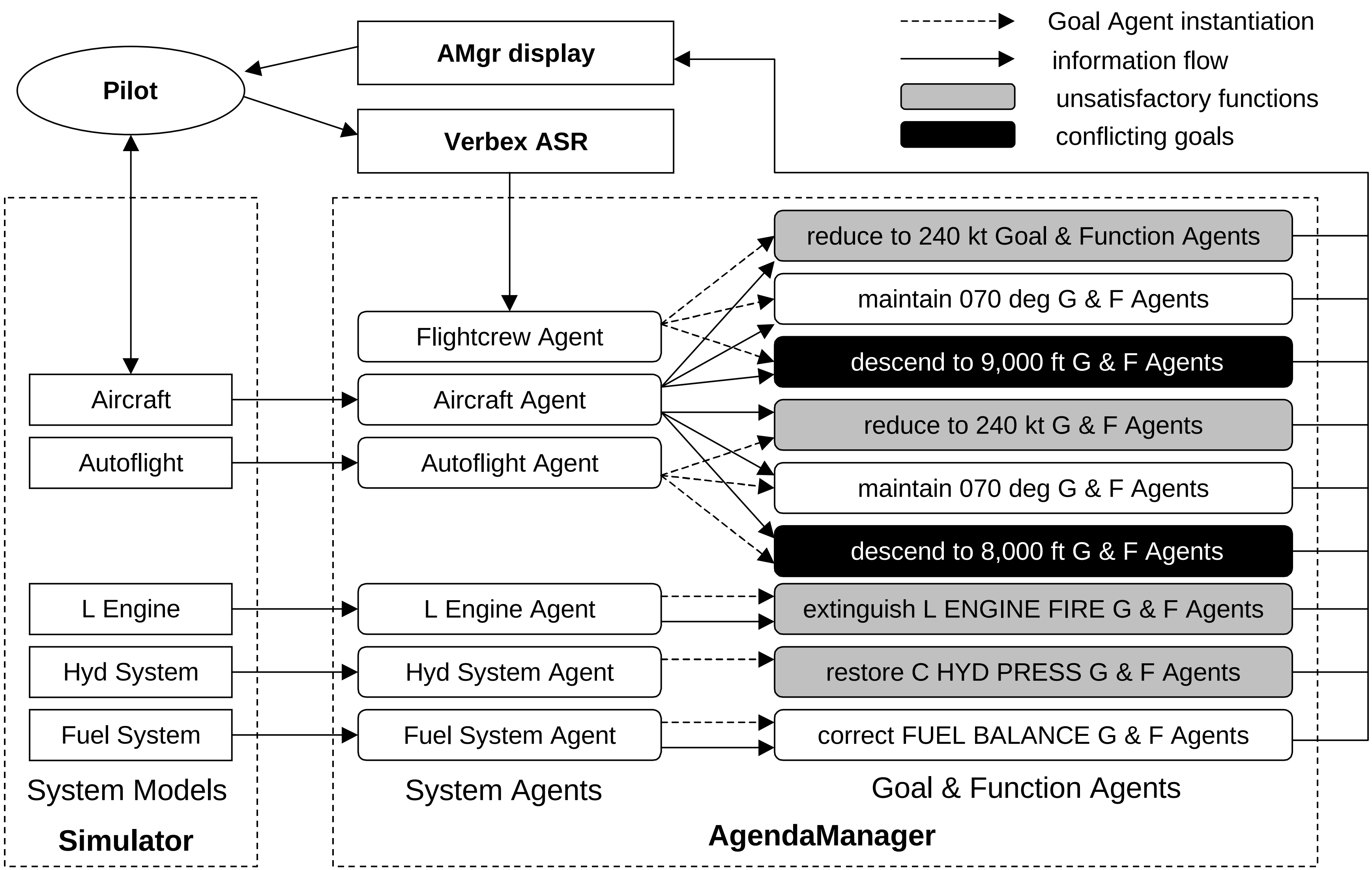

Figure 4. AgendaManager architecture.

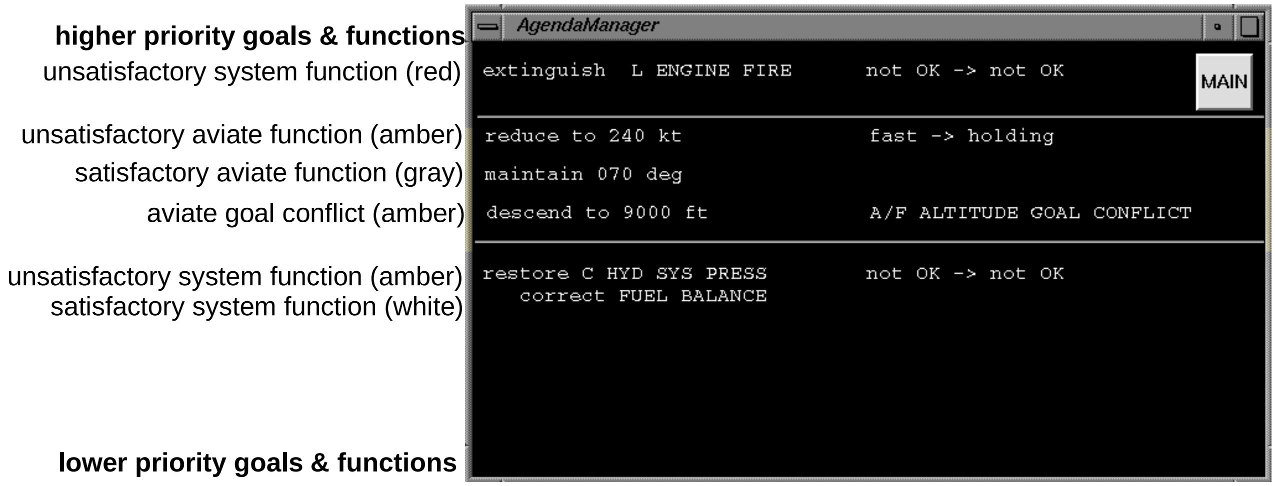

Figure 5. The AgendaManager display.

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 signaled 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.

Test and Evaluation – We conducted an experimental evaluation of the AMgr to determine its effectiveness in facilitating AMgt (again, a superset of CTM) as compared to 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.

The apparatus consisted of the following components

- a part-task flight simulator, running on two Silicon Graphics Indigo 2 workstations,
- the AgendaManager running on one of the two workstations,
- an experimenter's console running on a third workstation,
- and a Verbex VAT31 ASR system on a 486 personal computer, connected to the AMgr workstation by an RS-232 serial connection.

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 ASR 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:

- how correctly the subject prioritized concurrent subsystem functions;
- the average subsystem fault correction time;
- the average time to properly configure the autoflight system;
- the percentage of goal conflicts detected and corrected;
- the average time to resolve goal conflicts;
- how correctly the subject prioritized concurrent subsystem and aviate functions;
- the average number of unsatisfactory functions at any time;

- the percentage of time all functions were satisfactory; and
- the subject's rating of the effectiveness of each monitoring and alerting system: -5 (great hindrance) to +5 (great help).

The raw data for variables 1 - 8 were recorded by the AMgr itself. Goal Conflict objects recorded goal conflicts and Function Agents, which assessed function status as part of their roles, recorded function performance data.

Results – The data were analyzed using Analysis of Variance and Table 1 summarizes the results obtained for each of these variables.

Table 1. AgendaManager evaluation results: mean values (all times in seconds), p-values, and levels of statistical significance of the differences.

Response variable	AMgr	EICAS	p-value	level of significance
within subsystem correct prioritization	100%	100%	NA	not significant
subsystem fault correction time	19.5	19.6	.9809	not significant
autoflight system programming time	7.0	5.9	.1399	not significant
goal conflicts corrected percentage	100%	70%	.0572	0.10
goal conflict resolution time	34.7	53.6	.0821	0.10
subsystem/ aviate correct prioritization	72%	46%	.0308	0.05
average number of unsatisfactory functions	0.64	0.85	.0466	0.05
percentage of time all functions satisfactory	65%	52%	.0254	0.05
subject effectiveness rating (-5 to 5)	4.8	2.5	.0006	0.05

The first three variables, within subsystem correct prioritization, subsystem fault correction time, and autoflight programming time, showed no statistically significant 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 did not differ between the two conditions. Also, once a subsystem fault was detected, the process of correcting it was identical between the two conditions. Programming the autoflight system was 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 was 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 reflected this, and the results indicated how successful the AMgr condition achieved it (suggestive evidence of differences, with $0.05 < p < 0.10$). 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 (a statistically significant difference, with $p < 0.05$). 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; a statistically significant difference) by making more time available to them.

It is crucial for a pilot to recognize that primary flight control functions (i.e., aviate functions) are usually more critical than subsystem related functions (i.e., manage system 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% (a statistically significant difference). Last, but not least, with the AMgr the subjects were able to achieve a significantly higher percentage of time where all functions were performed satisfactorily (AMgr: 65%; EICAS: 52%; a statistically significant difference).

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. 2.5, a statistically significant difference).

Discussion of Results – 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 flew two-pilot aircraft. 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 ASR 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 as a superset of CTM is significant to flight safety, can be enhanced by means of a computational aid.

AGENDAMANAGER IMPLEMENTATION – Especially in light of recent advances in ASR technology and the Federal Aviation Administration's plans to introduce datalink technology to deliver ATC clearances to aircraft, we believe that further development of the AMgr is warranted. However, the AMgr remains a research tool and we are not currently implementing it for use in any real aircraft.

ON-GOING AND FUTURE RESEARCH – Our current research has narrowed again to CTM (a subset of AMgt) and is focussed on developing a better understanding it. The preliminary results of two of these studies provide additional support for our hypotheses that CTM (and therefore AMgt) should and can be improved. The first, a part-task simulator study conducted to investigate the effect of automation level on CTM performance, elicited CTM errors from airline pilot subjects, showing that CTM errors are frequent enough to warrant remedies. The second study, designed to discover the factors that influence the CTM process, suggests that task status -- how well the pilot *perceives* that he/she is performing the task -- is a major factor. Since one of the major functions of the AMgr was to monitor and report function/task status, this is an indication that the AMgr could be effective under real flight conditions.

These and our other results suggest several possible roles for the AMgr or its derivatives in the future. First, the AMgr has great potential as a CTM research tool. Most of our experiments so far have relied on laborious manual data reduction and analysis to measure CTM performance. But the AMgr assesses CTM performance in real-time. We plan to use a future version of the AMgr as a data collection tool that will monitor CTM performance during an experiment in real-time, allowing the experimenter to guide the scenario toward situations that will challenge CTM performance. Also, this will make CTM performance data available immediately after the experiment.

Second, the AMgr may be useful as a training tool. By monitoring CTM performance during simulated flight scenarios, a student pilot can be informed of poor CTM performance during and/or immediately following the training session. We hypothesize that knowledge of one's own CTM deficiencies would be an important step toward overcoming them.

Finally, we hope to further develop the AMgr to facilitate single-pilot instrument flight rules (IFR) operations. Single-pilot IFR is notorious for its challenges and dangers. The presence of an AMgr-like system to call the busy pilot's attention to unsatisfactory tasks may be a great safety enhancement.

CONCLUSIONS

There are three major conclusions to draw from our work. First, Cockpit Task Management is a significant factor in flight safety. Second, CTM can be facilitated, as was demonstrated by the AgendaManager.

But more important than these conclusions is the third. To successfully solve problems and realize opportunities in the air transportation system, technology must be developed using a needs-driven rather than a technology-driven approach. Many technologies, including artificial intelligence technologies, have great potential to improve the safety and effectiveness of air transportation, but they must be developed through a process that first identifies the needs and problems, then applies technology appropriate to those needs and problems. We believe that the human factors research and development approach described and illustrated in this paper is just such a process.

ACKNOWLEDGMENTS

The AgendaManager was developed under NASA Ames Research Center grant NAG 2-875. Kevin Corker and Barbara Kanki were our technical monitors. Continuing CTM research is being conducted under NASA Ames Research Center grant NAG 2-1287. Key Dismukes is our technical monitor. We greatly appreciate the financial support from NASA and the encouragement and wise counsel of our technical monitors.

REFERENCES

1. Anonymous (1979). *Military Specification: Human Engineering Requirements for Military Systems, Equipment, and Facilities*, MIL-H-46855B, Redstone Arsenal, AL: US Army Missile R&D Command.
2. Anonymous (1987). *Military Handbook: Human Engineering Procedures Guide*, DOD-HDBK-763, Washington: Department of Defense.

3. 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.
4. Boeing Commercial Airplane Group (1998). *Statistical summary of commercial jet airplane accidents, worldwide operations, 1959-1997*. Seattle, WA: Boeing Commercial Airplane Group, Airplane Safety Engineering.
5. 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.
6. Booher, H.R., Ed. (1990). *MANPRINT: An Approach to Systems Integration*. New York: Van Nostrand Reinhold.
7. Chou, C., Madhaven, D., & Funk, K. (1995). Studies of cockpit task management errors. *International Journal of Aviation Psychology 6*(4), pp. 307-320.
8. Chapanis, A. (1996). *Human Factors In Systems Engineering*. New York: John Wiley & Sons.
9. Damos, D. (1997) *The Effect of Operational Interruptions on Flight Crew Performance*. Los Angeles, CA: Damos Research Associates, Inc.
10. Flight Safety Foundation (1998). Aviation Safety: U.S. Efforts to Implement Flight Operational Quality Assurance Programs. *Flight Safety Digest*, July-September.
11. 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.
12. Latorella, K.A. (1996). Investigating interruptions: an example from the flightdeck, in *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*, Philadelphia, PA, 1996, pp. 249-256.
13. NTSB. (1973). *Aircraft accident report. Eastern Airlines, Inc., L-1011, N310EA, Miami, Florida, December 29, 1972* (NTSB-AAR-73-14). Washington: National Transportation Safety Board.
14. Rogers, W.H. (1996). Flight deck task management: a cognitive engineering analysis, in *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*, Philadelphia, PA, 1996, pp. 239-243.
15. SAE (1995). *Aerospace Recommended Practice: Pilot -System Integration*, ARP 4033, Warrendale, PA: SAE International.
16. Schutte, P.C., and Trujillo, A.C. (1996). Flight crew task management in non-normal situations, in *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*, Philadelphia, PA, 1996, pp. 244-248.
17. Weiss, G., Ed. (1999). *Multiagent Systems*, Cambridge, MA: The MIT Press.
18. Wilson, R.B. (1997). *Development and Comparison of Two Alternate Display Formats for an AgendaManager Interface*, MS Thesis, Corvallis, OR: Oregon State University.
19. Wilson, J.R. and Funk, K. (1998). The effect of automation on the frequency of task prioritization errors on commercial aircraft flight decks: an ASRS incident report study, in *Proceedings of the Second Workshop on Human Error, Safety, and System Development*, Seattle, Washington, USA, 1-2 April 1998, pp. 6-16.

CONTACT

Ken Funk may be contacted by writing to him at the Department of Industrial and Manufacturing Engineering, 118 Covell Hall, Oregon State University, Corvallis, OR 97331-2407 or by sending e-mail to funkk@ engr.orst.edu.