

항공시스템 아젠다 관리를 위한 에이전트 모델의 설계 및 평가

(Design and Evaluation of an Agent-based Intelligent System Modeling Architecture for Cockpit Agenda Management)

차 우 창[†]
(Woo Chang Cha)

요 약 컴퓨터가 장착되어 고도의 자동화된, 혹은 지능화된 비행기 조종실(cockpit)과 같이 기계와 그 기계를 운영하는 인간 작업자로 구성되어 있는 인간 기계 시스템에서, 안전하고 효과적인 시스템 수행을 위해 두 구성요소(기계와 인간)는 상대의 의도와 목적에 대해 충분한 지식과 정보를 공유해야 한다. 기계 인간 상호간의 의도와 목적을 정확히 파악하지 못하거나 서로의 목적이 상충되어 일어나는 항공 사고들이 종종 발생되고 있다. 인간과 기계 상호간의 의도와 목적을 정확히 파악하고 그 목적들의 수행을 체계적으로 관리하는 기술의 연구는 미미하다. 인간과 지능화된 기계 상호간 의도하는 바를 서로가 보다 정확히 인식하고 그 목적의 수행을 체계적으로 파악하는 기술이 요구되고 계속 축적되지 않는 한, 더욱 더 복잡해져 가는 인간 기계 시스템의 안전하고 효율적인 운영방안을 내놓을 수 없다. 이를 위해 한 방안으로 본 논문은 상용 비행기의 조종실 운영시 인간 파일럿과 기계 파일럿의 목적(Goal)과 그 목적을 수행하기 위한 직무(Function)를 통합 관리하는 아젠다(Agenda)의 관리(Agenda Management)의 에이전트 기반 설계 및 그 평가를 다루고 있다.

Abstract The pilot (human actor) involved in the control loop of the highly automated aircraft systems (machine actor) must be able to monitor these systems just as the machine actor must also be able to monitor the human actor. For its safety and better performance of the human machine system, each of the two elements must be knowledgeable about the other's intentions or goals. In fact, several recent accidents occurred due to goal conflicts between human and machines in a modern avionic system. To facilitate the coordination of these actors, a computational aid was developed. The aid, which operates in a part-task simulator environment, attempts to facilitate the management of the goals and functions being performed to accomplish them. To provide an accurate knowledge of both actors' goals and their function statuses, the aid uses agent-based objects representing the elements of the cockpit operations. This paper describes the development of the flightdeck goals and functions called Agenda Management.

1. 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 in one of the following categories:

- a. *aviate*: controlling the airplane
- b. *navigate*: determining where the airplane is, where to go, and how to get there
- c. *communicate*: communicating with air traffic control
- d. *manage systems*: configuring and correcting fuel, electrical and hydraulic systems

[†] 정 회 원 : 금오공과대학교 산업시스템공학과 교수

chaw@knut.kumoh.ac.kr

논문접수 : 1999년 9월 2일

심사완료 : 2000년 4월 3일

But in other cases, pilot errors occurred in *managing* flightdeck activities. The ordering of activities represented by the above list reflects a general ordering of activities by importance, and flightcrews attempt to allocate their attention to multiple, competing activities consistently with that ordering. But they occasionally fail to do so, sometimes with catastrophic results. The process of managing activities are called **Agenda Management** (hereinafter **AM**). AM is built on the assumption that on the modern aircraft flight deck (or, in the cockpit) human actors (pilots) cooperate with machine actors (autopilot and flight management system) to perform functions to achieve goals to control the airplane and its subsystem. This paper describes the development and evaluation of a prototype computational aid to facilitate AM, the management of flightdeck goals and functions.

2. Agenda Management

The concept of AM is an extension of a theory of Cockpit Task Management proposed by Funk [6]. 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, AM 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 (FMS). 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 climb to an altitude of 9,000ft, 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 climb to 9,000ft 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.

AM is a high-level flightdeck function performed cooperatively by flightdeck actors which involves two sub-functions:

- A. **Goal management** is the process of
 - A1. recognizing or inferring the goals of all flightdeck actors;
 - A2. canceling goals that have been achieved or are no longer relevant;
 - A3. identifying and resolving conflicts between goals; and
 - A4. prioritizing goal consistently with safe and effective aircraft operation.
- B. **Function management** is the process of
 - B1. initiating functions to achieve goals;
 - B2. assigning actors to perform functions;
 - B3. assessing the status of each function (whether or not it is being performed satisfactorily and on time);
 - B4. prioritizing those functions based on goal priority and function status; and
 - B5. allocating resources to be used to perform functions based on function priority

At any point in time, AM 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 considering only the management of functions performed by human actors (that is, task management [5]), 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 person. In an American Airlines Boeing757 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.

3. Research Objectives

From these preliminary findings it concluded that AM --and specifically the failure to perform AM satisfactorily--is a significant factor in flight safety. The objectives of this research was to develop and evaluate an experimental computational aid to facilitate AM. This aid was called the Agenda-Manager.

4. The AgendaManager

4.1 Simulator Environment

The part-task simulator models a generic, twin engine transport aircraft. It was 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 (our lab). The software was written in C, FORTRAN, and Smalltalk (VisualWorks 2.5).

4.2 Analysis and Design

As a first step in designing the AgendaManager, it was developed a formal, functional model of Agenda Management using IDEF0, a graphical modeling methodology useful for representing and decomposing complex activities [7]. 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 a data dictionary was generated consisting of the entities that are the inputs, outputs, and controls of the activities in the model. These were used to define the object-oriented architecture of the AgendaManager.

4.3 AgendaManager Architecture

Major AgendaManager objects include System Agents, Actor Agents, Goal Agent, Function Agent, an Agenda Agent, and an AgendaManager 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 AM. An Agent's declarative knowledge is represented using instance variables. Its procedural knowledge is represented using Smalltalk methods. The AgendaManager model and interface was developed on an object-oriented programming environment (specifically, *VisualWorks 2.51*, a visual

version of Smalltalk). 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 AgendaManager 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 Ethernet socket. An SA's declarative knowledge includes the past, current, and projected future state of 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 pastsystem 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.

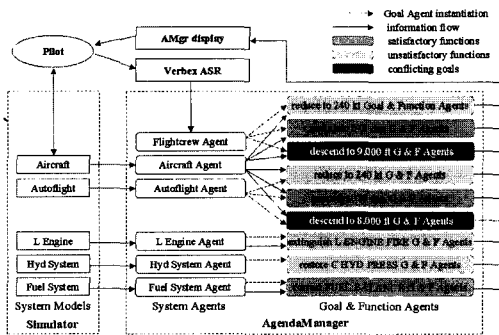


Fig. 1 AgendaManager Architecture

Actor Agents (AAs) recognize actors' goals, implicitly and explicitly, and make them known to the rest of the AgendaManager. 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 **Pilot** (Flightcrew)

Agent. The Pilot Agent (PA) is to recognize and declare implicit and/or explicit pilot intentions using two methods. One is to use a automatic speech recognition (ASR) system, which is called overt GCM (Goal Communication Method). The other is to infer the pilot goals using a script-based intent inferencing mechanism, so called covert GCM [3]. The PA has a serial connection to a Verbex ASR system. 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 PA, 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 PA which translates it and declares a goal by creating an instance of a Goal Agent.

To overcome workload concern when ASR used, the PA employs model-based implicit method to recognize pilot goals. Given the current state and flight phase, the PA tries to interpret pilot actions based upon script-based reasoning process depicted in Figure 2. If the action can be explained by an active script, corresponding active goal is recognized

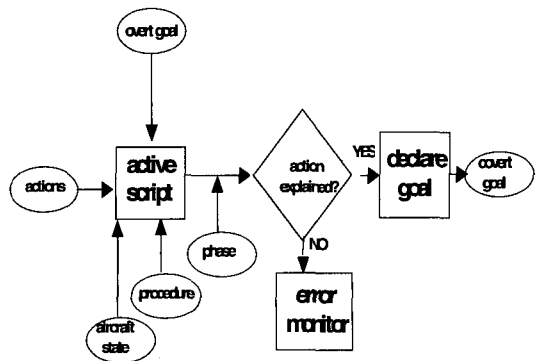


Fig. 2 Script-based reasoning process

by the intent inferencer, which represented a process model using a blackboard problem solving method. The knowledge source in this blackboard framework consists of a rule-based representation of goals and corresponding scripts. The recognized goals are declared by creating an instance of a Goal Agent [7].

An example Smalltalk code of active script is as follows:

```

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

```

The purpose of **Goal Agent** (GA) is to maintain information about all actors' goals. A GA represents an actor's goal, such as one to descent to, and to maintain an altitude of 9,000ft 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 Agent. 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.

4.4 Operation

As the simulator runs it sends state data to the AgendaManager, 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.

4.5 Pilot Interface

The **AgendaManager Interface** (AMI) consists of display formats for presenting agenda information to the pilot. It is illustrated in Figure 3, 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 AgendaManager

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 9,000ft.

He/she has verbally acknowledged this clearance and the AgendaManager has recognized these aviate(aircraft control) goals and instantiated GAs and FAs. Speed is currently too high and is not decreasing, so the AgendaManager speed message is amber and its comment notes the problem. The airplane's current heading is 070 degrees, so the AgendaManager's message for this is gray, with no explanatory comments, so as not to distract.

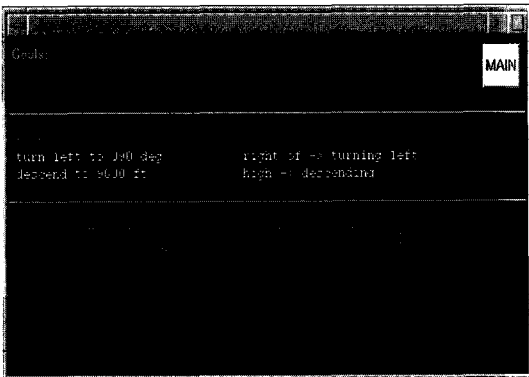


Fig. 3 AgendaManager Interface

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.

5. AgendaManager Evaluation

5.1 Objective

The purpose of the experiment was to determine any differences in AM performance between the use of the AgendaManager 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) shown at Figure 4. The EICAS interface that we used is the same image of EICAS display which is used in most commercial airplanes. 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.

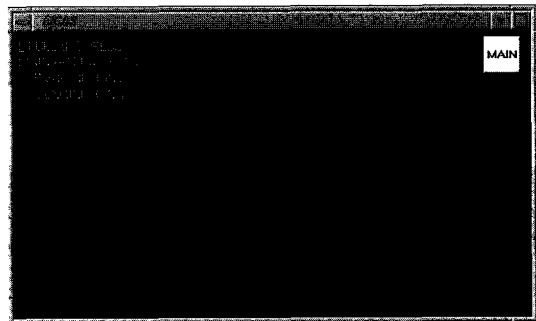


Fig. 4 EICAS Interface

5.2 Method

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 subjects learned how they 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 Agenda-Manager displays, and alter programmed flightpaths. After a lunch break, the subject flew two 30 minute

scenarios (one with EICAS, one with the AgendaManager), 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 AgendaManager 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)

5.3 Results

Table 1 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 AgendaManager/EICAS conditions. This is critical for the interpretation of the results in that it supports the hypothesis of the AgendaManager 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 AgendaManager 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 AgendaManager condition achieved it. Any time a goal conflict existed, the AgendaManager helped the subject identify this conflict (100%) whereas with EICAS, the subjects only identified 70% of the conflicts. Also, with the AgendaManager 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 (AgendaManager :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

Table 1 AgendaManager evaluation results, mean values (all times in seconds)

Response variables	AgendaManager	EICAS	p-value
within subsystem correct prioritization	100%	100%	n/a
subsystem fault correction time	19.5	19.6	.9809
autoflight system programming time	7.0	5.9	.1399
goal conflicts corrected percentage	100%	70%	.572
goal conflict resolution time	34.7	53.6	.821
subsystem/aviate correct prioritization	72%	46%	.308
average number of unsatisfactory functions	0.64	0.85	.466
percentage of time all functions satisfactory	65%	52%	.254
subject effectiveness rating (-5 to 5)	4.8	2.5	.006

functions. The AgendaManager 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 AgendaManager the subjects were able to achieve a significantly higher percentage where all functions were performed satisfactorily (AgendaManager: 65%; EICAS: 52%). 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 AgendaManager (4.8 vs 2.5).

6. 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 AgendaManager 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 [7]. Given the industry's objective of significantly reducing the number of commercial transport accidents, the AgendaManager must be seen as one of the facilitating tools in this effort.

Based on our results, we believe that there are several research paths to be explored. For example, the AgendaManager 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 AgendaManager development, we experimented with a goal communication method that integrated overt communication (via clearance acknowledgement) and covert communication (via script-based intent inferencing)[4]. Although we chose to focus on overt goal communication in the current version of the AgendaManager, 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 AgendaManager 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 assessment. 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 AgendaManager has potential as an operational aid, its near-term benefits may be realized in other ways. For example, with suitable modifications, the AgendaManager could be embedded in a part-task trainer to facilitate AM training. Another possible role is as a research tool. With relatively minor changes the AgendaManager could be used to capture AM data on-line in full-mission simulator experiments. In fact, the greatest value of the AgendaManager may be in this capacity, helping us understand the phenomenon of Agenda Management better.

7. Acknowledgements

This research was partly supported in 1999 by Research Fund, Kumoh National University of Technology, modified 95-97 research funded by NASA Ames Research Center.

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. "Developing an Embedded Method to Recognize Human Pilot Intentions In an Intelligent Cockpit Aids for the Pilot Decision Support System," *Journal of the Ergonomics Society of Korea*, Vol. 17, No. 3, 1998
- [4] Cha, W.C & Funk, K., "Communication 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.
- [5] Chou, C.D., Madhavan, D., & Funk, K., "Studies of Cockpit Task Management Errors," *International Journal of Aviation Psychology*, 6(4), 1996, pp.307-320
- [6] 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
- [7] Funk, K. & Braune, R., "Expanding the Functionality of Existing Airframe Systems Monitors: The AgendaManager," *Proceedings of the Psychology*, 27 April - 1 May 1997, Columbus, Ohio.



차 우 창

1984년 한양대학교(공학사-산업공학).
 1984년 ~ 1987년 현대전자 정보사업부.
 1990년 Ohio Univ.(공학석사-산업공학).
 1996년 Oregon State Univ.(공학박사-인간공학). 1997년 Oregon State Univ. 연구교수. 1998년 ~ 현재 금오공과대학교 산업시스템공학과 조교수. 관심분야는 인지공학 및 항공 안전, 인간성능 모델링 및 시뮬레이션, HCI