AN ABSTRACT OF THE THESIS OF

Woo Chang Cha for the degree of Doctor of Philosophy in Industrial Engineering

presented on October 7, 1996.

Title: Communicating Pilot Goals to an Intelligent Cockpit Aiding System.

Abstract approved: Redacted for Privacy

Kenneth H. Funk

A significant number of aircraft incidents and accidents have been caused, in part, by flightcrew failure to properly manage cockpit activities, such as failure to initiate activities at the appropriate time, misprioritization of activities, or the failure to appropriately monitor activities and terminate them when required. To facilitate the management of the cockpit activities, a computational aid, the Agenda Manager (AM) has been developed for use in simulated cockpit environments in an investigation which was one aspect of a more extensive research project supported by the NASA Ames Research Center.

The AM is directed at the management of goals and functions, the actors who perform those functions, and the resources used by these actors. Development of an earlier AM version, the Cockpit Task Management System (CTMS), demonstrated that it could be used to assist flightcrews in the improvement of cockpit activity management under experimental conditions, assuming that the AM determined pilot goals accurately as well as the functions performed to achieve those goals.

To overcome AM limitations based on that assumption, a pilot goal communication method (GCM) was developed to facilitate accurate recognition of pilot goals. Embedded within AM, the GCM was used to recognize pilot goals and to declare

them to the AM. Two approaches to the recognition of pilots goals were considered: (1) The use of an Automatic Speech Recognition (ASR) system to recognize overtly or explicitly declared pilot goals, and (2) inference of covertly or implicitly declared pilot goals via use of an intent inferencing mechanism. These two methods were integrated into the AM to provide a rich environment for the study of human-machine interactions in the supervisory control of complex dynamic systems. Through simulated flight environment experimentation, the proposed GCM has demonstrated its capability to accurately recognize pilot goals and to handle incorrectly declared goals, and was validated in terms of subjective workload and pilot flight control performance.

Communicating Pilot Goals to an Intelligent Cockpit Aiding System

by

Woo Chang Cha

A Thesis

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed October 7, 1996
Commencement June 1997

Doctor of Philosophy thesis of Woo Chang Cha presented on October 7, 1996

APPROVED:

Redacted for Privacy

Major Professor, representing Industrial Engineering

Redacted for Privacy

Head of Department of Industrial & Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Woo Chang Cha, Author

ACKNOWLEDGMENTS

This research was partly supported by National Aeronautics and Space Administration (NASA) research grant NAG 2-875 from the Ames Research Center to Oregon State University. The technical monitor was Kevin Corker.

First and foremost, I am grateful to my wife, Kyoung-Mi, and our children, Kyoung-Hoon and Ji-Hoon, for their love, support, and encouragement throughout this long course of study.

Next, I would like to thank my advisor, Ken Funk, for his spiritual as well as academic guidance. He has touched my soul and spirit deeply. I was very glad to find that we share goals and beliefs about the study of human factors engineering as a means to compensate for human fallibility.

I also wish to thank to my committee members, Drs. Bruce D'Ambrosio, Sabah Randhawa, Brian Paul, and. Eugene Abrassart, who helped me to complete this study through their comments, suggestions and encouragement. Special thanks to the technical editor, Bill McMechan, who helped me to convey my thoughts into English effectively, and our flight simulation domain expert, Rolf Braune. He helped me to understand flight domain more accurately and to thus develop a more reliable and realistic simulation environment. Many thanks to the subjects who voluntarily participated in the experiment for this study and to Robert Wilson, who is a sincere companion, sharing thoughts about our studies and even our spiritual lives.

Finally, thanks to my Lord, Jesus Christ. You gave me many chances to know You through this study. Sometimes I wondered where you were, but now I know you were always there in my heart and in my life. I wish to dedicate this dissertation to you, my Lord, and to your earthly servant, the Corvallis Korean Church. I would have you use my heart and knowledge as a tool to re-build this earth more perfectly and beautifully than it presently is. *Maranatha.*

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

Communicating Pilot Goals to an Intelligent Cockpit Aiding System

1. INTRODUCTION

Air travel is one of the safest forms of transportation statistically. In fact the commercial aircraft accident rate has continued to decline over the past several decades. Nonetheless, when air accidents do occur , enormous losses of both life and property are occasioned. Pilot error is responsible for approximately two-thirds of all aircraft accidents. Procedural errors are also frequently involved (Nagel, 1988; Hawkins, 1993; Boeing Commercial Airplane Group, 1994).

Modern aircraft systems are equipped with sophisticated and automated cockpit controls based upon advanced technologies, the intent of which is to ensure system reliability and to improve aviation safety. Cockpit automation has improved flight safety by reducing pilot and procedural errors and in the modern era, the air accidents that occur are often due to a totally different category of pilot error, namely task management errors. The pilot's changing role in monitoring cockpits has led to this new type of error, cockpit task management (CTM) errors, which occur during the initiation, prioritization, monitoring, and termination of cockpit tasks (Chou & Funk, 1990; Chou, 1991; Funk, 1991; Madhavan, 1993).

With anticipated increases in the amount of air travel as well as the complexity of aircraft systems, it is essential that accurate aids be developed to help pilots manage and perform cockpit tasks. A primitive aid developed to facilitate pilot CTM performance, the cockpit task-management system or CTMS, demonstrated an effective means for the improvement of CTM under experimental conditions. This system was based upon the assumption that an aid system can be aware of those tasks the pilot seeks to perform and how well they are subsequently performed (Kim, 1994). However, a task management aid, such as the CTMS poses several limitations when applied to realistic situations. One of its critical limits is the lack of a reliable means of communication to ensure that the aid is accurately aware of the pilot's intentions or goals (Callantine & Mitchell, 1994;

Hoshstrasser & Geddes, 1989; Skidmore et al., In press). In fact, the miscommunication of goals between human operators and mechanized actors brought about a critical situation contributing to the crash of a China Airlines A300-600R at Nagoya, Japan (*Aviation Week & Space Technology*, 1994; 1996).

The purpose of this study was to examine methods for recognizing pilot cockpit operational goals. The research objective was thus to develop an accurate methodology through which a pilot can communicate intentions or goals to an associated intelligent-aid system, and to determine its effects upon cockpit automation, contributing to the proposed integration of such communication methods into an enhanced CTM aid for real-time flight simulation environments and ultimately, real cockpits.

This dissertation is organized as follows. Chapter 2 presents background information necessary to understand the present project, including a study of conventional and computational aids, the nature of CTM errors, and current approaches to the improvement of CTM systems as well as an awareness of their limitations. Chapter 3 outlines the objectives of this research. Chapter 4 presents a description of the methodology used to conduct the research, including the use of a flight simulator, implicit and explicit methods for communicating goals, and an evaluation of the method. Chapter 5 provides a description and analysis of the experimental results and a discussion of these results and research conclusions are presented in Chapter 6. Finally, in Chapter 7 the potential contributions and limitations of this research to aviation safety and other domains are discussed.

2. BACKGROUND

2.1 Procedural Error and Flight Safety

So long as humans are involved in aircraft operations as pilots, system designers must consider the contribution of related human errors to flight safety in their systems. In point of fact, approximately two-thirds of all air accidents are attributable to errors committed by the cockpit crew (Boeing Commercial Airplane Group, 1996; Nagel, 1988), including a number of accidents related to procedural errors committed by human pilots. In a study of 93 major aircraft accidents during the period 1953 through 1983, Sears (1986) determined that approximately 33% of the accidents involved procedural errors. In several recent accidents, improper use of flightdeck checklists has been cited as a factor in several recent accidents (NTSB 1988, 1989, 1990). In research sponsored by the NASA Ames Research Center, Degani and Wiener (1990, 1991) found procedural compliance to be a significant factor in the maintenance of aviation safety.

Two efforts have been undertaken to reduce cockpit procedural errors. One has focused upon the conventional aids used in actual operations (i.e., training systems or improved checklists), and the second direction has involved computational aids developed in laboratory simulation studies. Examples include such artificial intelligence (AI) applications as the Pilot's Associate (PA) developed by Rouse and colleagues (Rouse et al., 1987, 1990) or the cockpit assistant system, CASSY, developed at the University of the German Armed Forces, Munich (Onken & Prevot, 1994; Gerlach et al., 1995). Pilot training and improved checklists will no doubt serve to improve procedural compliance in actual cockpit operations, but there are also serious problems with this type of conventional approach. First, training is expensive and is capable of accomplishing only limited goals as the complexity of aircraft systems increases rapidly. Second, pilots are required by law to use checklists. There are, however, problems with checklists if the individual pilot chooses not to use them for any reason (Degani & Wiener, 1991; Roth, Bennett, & Woods, 1987).

As computer technologies have been developed, computational aids based upon techniques of AI provide another feasible alternative approach to the reduction of procedural errors in complex and dynamic avionics systems. Funk and Lind (1992) used a distributed AI methodology to develop a task support system (TSS) capable of helping pilots plan, manage, and execute mission-related tasks. Hammer (1984) developed a rule-based system for intelligent flight management of procedural execution. In turn, the PA or Pilot's Associate is a collection of AI systems that help fighter pilots in tasks required for air-to-air combat (Rouse et al., 1987, 1990; Banks & Lizza, 1991). While AI applications have seemed to work well in simulator-based evaluations in laboratory settings (Geddes & Hammer, 1991), their effectiveness for use in real-world avionics systems is subject to question. Since these aids use a pilot model-based approach, they may be unrealistic for near-term research and development because of the limitations of current cognitive models. Moreover, use of the current generations of procedural aids to automate pilot skills have their own limitations when applied to realistic situations, posing critical weaknesses caused by the creation of a new types of workload based upon the need to monitor automated procedures, and by new types of errors, known as cockpit task management (CTM) errors.

2.2 Automation and Cockpit Task Management Errors

During the 1980s, cockpit automation was regarded within the aviation industries as the savior of competitiveness and was often over-used in many areas as reliable computer technologies were developed. Increased automation has brought about major changes in the pilot's role as controller of complex, dynamic aircraft systems. In particular, the pilot's role has shifted from being that of a manual controller, for which perceptual-motor skills are emphasized, to being a supervisory controller, for which such cognitive skills as planning, monitoring, and decision making are emphasized (Rasmussen, 1986; Rouse, 1981; Wickens, 1992).

The role of pilot as the supervisory controller of a complex, dynamic avionics system often leads to new cockpit problems, including (1) an increased monitoring workload, (2) over-reliance on automation to such an extent that any pilot intervention is seemingly unnecessary, and (3) slow and inappropriate responses to system failures by a supervisory controller acting primarily as a passive monitor rather than as an active controller (Wickens, 1992; Billings, 1991). These problems indicate the necessity of dealing with the new type of pilot errors that occur in connection with monitoring and/or managing cockpit tasks. In fact, in modern, highly automated aircraft, pilots now may make a greater number of errors performing managerial tasks than they do when performing control tasks (Chou, 1991; Chou and Funk, 1993, Madhavan, 1994; Sarter and Woods, 1992).

Flightcrew management-level tasks include monitoring system configurations and making decisions, as well as performing low-level tasks specified by checklists and other procedures. Funk (1991) referred to the management-level activity as cockpit task management or CTM, viewed as "a process by which the flightcrew manages an agenda of cockpit tasks." Procedures were thus developed to describe CTM activities, including the initiation, monitoring, prioritization, and termination of multiple, concurrent cockpit tasks. Subsequently, Chou and Funk (1990) developed a taxonomy of CTM errors from an analysis of 324 National Transportation Safety Board aircraft accident reports, and Chou (1991) asserted that many air accidents can be explained as failures by the flightcrew to perform CTM functions correctly. It was determined that 77 (23%) of the accidents reviewed involved significant CTM errors. Madhavan (1993) examined CTM errors based upon 470 Aviation Safety Reporting System (ASRS) incident reports, and reported that CTM errors were involved in almost 50% of the incidents reviewed.

In any event, the replacement of pilots with automated procedures is not a feasible response to the need to reduce CTM-like errors. Human pilots remain far superior to computer technology in their ability to deal with significant quantities of uncertain variables, including the weather, operational constraints, and related ground situations (Nagel, 1988; Wickens, 1992; Reason, 1990). On the other hand, in a complex and dynamic avionics system, it is almost impossible for a pilot to anticipate and plan procedures to confront all possible flight contingencies (Rasmussen, 1986; Wickens, 1992). Rather, advanced automation (i.e., computational technologies, such as an intelligent decision-support system) may be used to amplify rather than to automate human skills for the improvement of CTM (Funk and Lind 1991; Kim 1994).

2.3 An Approach to Improving Cockpit Task Management: The CTMS

Preliminary CTM analysis has shown that facilitating CTM performance and reducing CTM-related pilot errors provides a potential contribution to aviation safety. Kim (1994) performed a laboratory study of CTM behavior using the Cockpit Task Management System (CTMS) as an aid to facilitate CTM. The CTMS provided the flightcrew with information about task state (upcoming, active, terminated), status (satisfactory or unsatisfactory performance), and priority. He used concepts of object-oriented design and distributed artificial intelligence in the CTMS implementation, where aircraft subsystems and flight tasks were represented by intelligent software units referred to as agents. In the CTMS, aircraft subsystems and pilot tasks were represented by system agents (SAs) and task agents (TAs), respectively.

From his agent-based prototype aid, he tried to demonstrate that the aid would help flightcrew prioritize tasks, initiate tasks, terminate tasks, interrupt tasks, and resume interrupted tasks. The experimental results obtained indicated that use of the CTMS provided effective improvement of CTM performance under experimental conditions wherein preprogrammed pilot goals and tasks were implicitly known by the software. In other words, if an aid can know accurately which tasks the pilot seeks to carry out, and how well the tasks are subsequently performed, then CTM can be improved by informing the pilot of such relevant task management information as suggested task state, status and priorities.

If CTM is to be improved, it is important that such assistants be able to recognize pilot intentions or goals accurately. In the absence of accurate knowledge of pilot goals, the task management information provided by CTMS-like aids may communicate the wrong directions (e.g., misprioritization) to the pilot using the information, or not even know what the pilot tasks are. Using the CTMS, pilot goals were declared via low-level operations for certain events and the specific tasks incumbent upon these events (e.g., following a specific procedure such as moving to a lower altitude in the event of an engine failure). However, when several events occurred simultaneously, it was often impossible for the CTMS to know exactly what the pilot intended to do. Thus, due to inaccurate knowledge of pilot goals, CTMS task management information can provide incorrect directions to a pilot using the information.

To know or be aware of a pilot's exact goals in multiple and concurrent task environments, an appropriate CTMS-like aid mechanism should encompass the appropriate means to implicitly infer pilot intentions (Geddes, 1985; Geddes & Hammer, 1991), or a method for recognizing pilot goals by the explicit communication of such goals (Callantine and Mitchell, 1994). In fact, however, the CTMS was based upon only very limited methods for either the implicit or explicit communication of pilot goals. Therefore, an enhanced CTMS employing goal-communication methods is required for the improvement of CTM to the extent that certain of the benefits accruing from the use of the CTMS under laboratory conditions may as well be obtainable from use in more realistic environments.

2.4 Goal Communication Methods

2.4.1 Introduction

The pilot (hereinafter, the human actor) involved in the control loop of a cockpit automated system must be able to monitor these systems (hereinafter, the machine actors), just as the machine actor must also be able to monitor the human actor. Each of the two elements must be knowledgeable about the other's intentions or goals. Several intelligent procedural aids, such as the PA or the CTMS, have been developed for this cross- monitoring function as planning and situation assessors (Rouse, 1988; Kim, 1994).

However, it is often difficult for the human actor to efficiently describe the complete set of his interacting goals to the machine actor. That is, the human actor has an explanation problem with respect to the machine actor. In such a complex, dynamic domain as aviation, human ability to explain intentions to the intelligent system is highly constrained by both time and the expressive capabilities of a non-textual interface (Hammer, 1984; Hoshstrasser, 1991). Thus, achievement of human-to-machine goal communication is the main objective of this research.

2.4.2 Goal Communication

A goal represents the actor's intentions to achieve a desired system state or system behavior. Goal communication consists of goal-directed internal representations shared between pilots (human actors) and intelligent subsystems (machine actors) in overt (explicit) or covert (implicit) forms that both actors readily understand. To design a goal communication framework for the control of an avionics system, it is increasingly important and useful to distinguish between overt and covert channels of communication (Rouse, 1981). Using the overt channel, human actors are conscious of the fact that information is being sent or received. On the other hand, using covert channels, the sending or receipt of information is unconscious on the part of the human actor.

2.4.3 Overt Goal Communication

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

Although speech interactions between human and machine actors have resulted in system performance instabilities, voice interactions using Automatic Speech Recognition (ASR) technology have received increased attention as a means of direct and accurate overt goal communication. Despite the fact that current ASR technology has heavily focused on telecommunication areas such as voice activated telephone services, ASR has generally been considered a promising method for declaring pilot goals in a wide range of airborne environments, from helicopters and fast military jets (Mountford and North, 1980; Reed, 1985; Williamson et al., 1996) to civil aircraft (Starr, 1993). The application domain of flying an airplane is recognized as being potentially reluctant to the use of ASR, because it exhibits some factors that characterize adverse environments for ASR and may have negative effects on pilot performance, such as high noise levels, high acceleration forces, and extreme levels of workload and stress (Williamson et al., 1996; Baber, 1996). Nevertheless, research into the means to use ASR in the aviation domain is increasing simply because of its potential for reducing pilot workload. ASR permits "eye-and hands-free" interaction with flight control systems and allows pilots to maintain head-up flight with "hands on throttle and stick" control.

In fact, several studies have demonstrated that voice can be used as an implementation method to declare the goals of the pilot. A continuous-speech recognition system was successfully integrated into the Lockheed-Georgia electronic copilot for advanced tactical fighters (Aerospace, 1987). In this instance, the pilot can check his weapons by calling for a symbolic display of missile capabilities. Then, by voice or button, the pilot can select the symbol for the missile to indicate its readiness and a synthesized voice confirms the request. In addition, in the Wesson International air-traffic controller-simulator, the computer-simulated pilot confidently acknowledges a controller's spoken instructions and simulated airplane blips move accordingly on a radar scope (Smith, 1994). The successful flight tests of the Cockpit Assistant System CASSY in real Instrument Flight Rule (IFR) flights have demonstrated that it is possible to integrate into modern aircraft intelligent on-board systems which use ASR as pilot interface for pilots to declare low level goals or commands explicitly (Gerlach et al, 1995; Onken, 1994). And also Williamson and his colleagues at the Pilot-Vehicle Interface Branch of Wright Laboratory completed successful flight tests of a speaker-dependent, continuous speech recognition system onboard a NASA Lewis Research Center OV-10A aircraft, with an experimental result of 97 % average word recognition accuracy (Williamson et al., 1996).

Although many industries, including avionics, have attempted to use an ASR system in their real field, current voice technology limits its effectiveness (The limitations are discussed in 2.4.5.). Nevertheless, an ASR system was used as a method to declare explicit goals in the present cockpit task management study, simply because of the fact that pilots are consistently communicating their goals verbally with air-traffic controllers or cockpit partners, and the technologies are being developed with some rapidity.

An alternative to explicit communication of pilot goals is to simply select from a visually presented goal list menu consisting of every possible goal to be declared. By selecting an intended goal on the goal list using a "mouse" or other pointing device, the pilot can explicitly state intentions to a machine actor. Although this menu-driven communication method may not be practical, insofar as it cannot display all the goals that could possibly be selected, it may nonetheless provide a useful method for evaluating the protocol of one communication method in comparison to other proposed communication methods. Actually, this method was used in this research for development purposes to declare pilot goals when the ASR system was not available.

In addition, such pilot actions as gesturing or pointing to information on a CRT can provide a potential means of overt communication. McDonnell-Douglas Aircraft proposed "a cockpit-of-the-future," equipped with a viewing screen and an upper extension that would show the pilot where to fly. Using this system, use of a touch screen and voice commands would serve to replace a traditional instrument panel and controls (*Aerospace*, 1987). Eventually, the design objective is to create a system that will allow pilots to declare their flight goals by simply fixing their eyes upon visual symbols on the viewing screen. More advanced systems have also been envisioned to contain brain wave and eye movement sensors for monitoring the physical condition of the pilots.

Basically, all activities declaring a pilot's goals explicitly will be considered to be explicit goal communications, even should such communications imply covert communications. For example, if the pilot should push the flight level change switch on the MCP to the on position, the activity itself will represent an explicit goal communication since the pilot has explicitly declared the goal of changing the flight level. At the same time, such a goal would automatically imply the holding of current heading and to trigger vertical speed modes. Goals for the heading hold and vertical speed modes will be implicitly declared from the implicit goal communication method described in Section 2.4.5.

2.4.4 Limitations and Design Issues of Overt Goal Communication Methods

As briefly discussed in the previous section, an ASR system was employed in this research as an overt goal-communication method since pilots declare some of their goals verbally in a cockpit operation and voice technologies have been developed to an efficient state. Human pilots may declare their goal verbally by attempting to establish *habitable* subsets of a natural language like English, such that pilots can get used to speaking only the allowed wordings of the subset, while the ASR system can reliably handle inputs from the subset of allowed utterances, the issue of designing the *habitable* vocabularies is important to ensure that the vocabulary selected is appropriate to the user's perception of the task. Some the work at Unisys and the Speech Science Institute might prove useful in clarifying what defines habitability (Lea and Dahl, 1996).

Research has indicated that under time pressure, people tend to increase their speech rate (Baber, 1995). This does not present too many problems for ASR, provided the device used is capable of detecting the spaces between words in a speech stream (continuous speech recognition). Baber also suggests that a significant human factors problem arises from the fact that people have difficulty remembering correct command words when under pressure. Thus, a major problem related to ASR use involves the ability of users to recall correctly all vocabulary items. While some of the problems might be removed by training pilots in effective ways of speaking (Hapeshi and Jones, 1989), some stressful situations (such as multi-task cockpit operations) also impair the use of working memory, so that comprehensive user training appears to be impractical. The author is not aware of any undertakings that specifically address the effect of training on ASR use during stressful situations.

Some applications of ASR rely heavily on providing visual feedback to operators. In addition to the effects of acceleration on visual perception, research has shown that visual feedback in ASR is prone to be misleading (Baber et al., 1995). Adequate feedback to ASR must be not only easy to read but also displayed for sufficient time that the user does not have to rely on short-term memory. Given the potential problems of feedback monitoring, further research should focus on how to repeat or correct misinterpreted commands.

To ensure more successful use of this technology for studies such as the present one, Baber and his colleagues issued human-factors oriented system design guidelines for use in an adverse environment of avionics including vocabulary selection, dual-task performance, training, and voice feedback monitoring (Baber, 1996; Baber and Noyes, 1996).

2.4.5 Covert Goal Communication

A system is an object with inputs, outputs and states which change over time. The aircraft is a system which may be decomposed into subsystems including the human pilot, the autopilot, and the engines. In a dynamic aircraft system environment, the subsystem human actor (pilot) communicates with machine actors. In human-machine interactive communication, the machine actor senses the state of the system, transforms it, and presents it to the human actor (i.e., the pilot), who then performs the desired control tasks. This form of communication can be covert in the sense that the human need not be conscious of the mechanized transformation process. Consequently, this form of communication differs somewhat from overt communication. Particularly with respect to human-to-machine direct communication (which is the principal subject of this research), it is possible to conceive of the generation of mechanized control signals based upon the covert sensing of the state of a system through a human actor, but without human awareness of being used as a transducer.

There are two primary reasons for the employment of covert means of goal communication. The first reason is to avoid the workload associated with overt communication. For example, if the machine actor could be enabled to covertly assess the human actor's intentions, then the human would not be distracted from other activities for the purpose of supplying this information. The second is based upon the possibility that, at certain times or in certain situations, it will not be possible to communicate goals overtly. As an illustration, when human workload is at a high level, it is possible or even likely that the human actor may not be able to recognize an incoming stimulus; or, even if it is recognized, the human actor may not, at that point, wish to take the time necessary to further communicate intentions to the machine actor.

To communicate covertly or implicitly with an intelligent aid in a highly dynamic system, the human actor simply performs selected procedural steps from which a model-based intent inferencer can infer goals from the procedural actions (Gerlach et al., 1995; Onken and Prevot, 1994; Geddes, 1985, 1989; Mitchell, 1987; Rubin et al., 1988). In other words, covert communication models are embedded within an aid, from which they may subsequently be used to assess the human state with respect to how human performance compares with that predicted by the model. That is, the intent model interprets operator actions in the context of goals, plans, and scripts, which are loosely ordered sets of tasks and actions, and carries out the plans. In this sense, Geddes (1985, 1989) demonstrated a successful intent model for fighter aircraft pilots performing air-to-air combat missions.

The operator function model (OFM, Figure 2.1) is a prescriptive model that specifies nondeterministically a set of plausible operator functions, subfunctions, tasks, and actions, from the current state of the system and recent operator actions (Mitchell, 1987). It provides a flexible framework for the representation of operator functions as well as system knowledge, and is thus a mechanism that can be used to define expectations for operator activities, given the current state of such activities. In turn, the OFM expert system (OFMspert) was developed to explore the design of operator aids for complex dynamic systems, using the OFM to organize knowledge about operator intentions (Rubin et al., 1988).

The action interpreter (ACTIN) is the OFMspert component that is primarily responsible for dynamic intent inferencing. The ACTIN blackboard data structure represents hierarchically hypothesized, current, operator intentions as a decomposition of operator goals into related plans, tasks, and actions while implementing the OFM intent-inferencing function (Jones et al., 1988). These intent-inferencing capabilities have been validated and were shown to be comparable to those of domain experts for monitoring operator supervisory control actions (Jones et al., 1990). However, insofar as the results of validation testing for intent inferencing did show a small number of misinterpretations, any possibility of misunderstanding human actions could result in serious errors when applied to real situations.

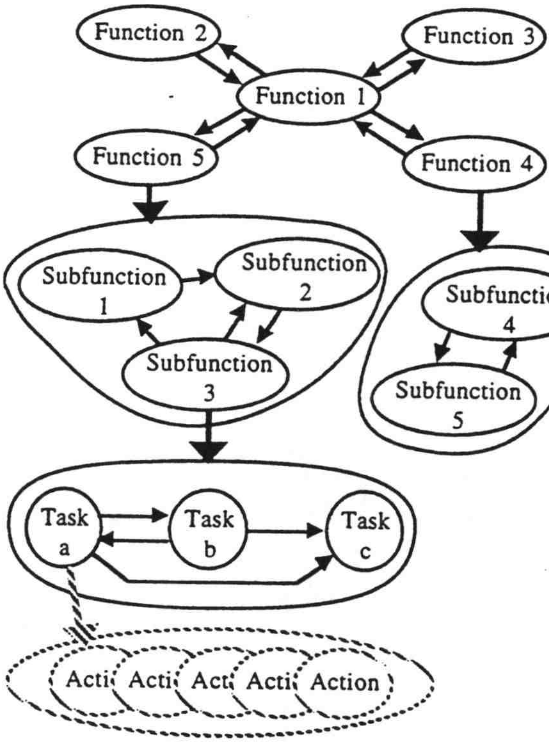

Figure 2.1 Generic Operator Function Model (OFM)
(source: Callantine, 1996)

Thus, an enhanced OFM architecture, the OFM-ACM, was developed and applied to aviation navigation for tracking activities performed by the crew of a "glass cockpit" aircraft while using automated flight modes to fly desired paths. This experimental system is referred to as the Georgia Tech crew-activity tracking system (GT-CATS) (Callantine and Mitchell, 1994). GT-CATS enhanced OFMspert by adding phase of operation knowledge in a goal hierarchy. It also decomposes operator function into automatic control modes, which can be used to perform the functions. Each mode in turn decomposes into the tasks, subtasks, and actions required to use it, depending on the situation. (Figure 2.2)

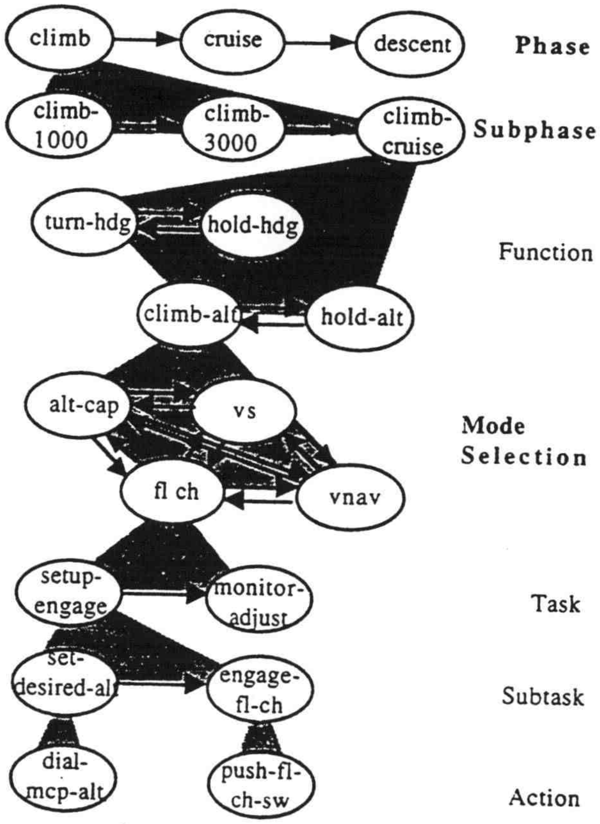

Figure 2.2 Portion of the GT-CATS OFM-ACM Showing the activity engaging Flight Level Change (FL CH) mode for climb (source: Callantine, 1996)

CASSY, or the Cockpit Assistant System, provides another means to infer pilot goals by use of a model for assessing situations (Onken, 1995; Gerlach et al., 1995). CASSY is a knowledge-based on-board assistant system with interfaces to the flightcrew, to air traffic control and to aircraft systems. In CASSY, the flight plan served as input for the *Piloting Expert* module in which, assuming the crew pursues the given flight plan, expected pilot actions were generated using a normative, individual model of pilot behavior. These expected actions were compared with the actual behavior of the flightcrew in the *Pilot Intent and Error Recognition* module. A successfully recognized intent served as new input and the pilot goal was declared to the corresponding interface. Although the flight test results of CASSY indicated 8% errors in inferring the pilot's intent, it demonstrated the availability of features like pilot intent recognition in a complex modern aircraft.

2.4.6 Limitations of Covert Goal Communication Methods

In summary, goal communication has been an important safety concern, increasing as the use of automation increases in modern aviation systems. Although the use of covert forms of human-machine communication remains a speculative interest, with experiments limited to the confines of the research laboratory, the development of covert applications is a particularly important task in this area of research. If this type of communication method is not made available, it is possible that the workload for human actors associated with overt communications may eventually counteract the benefits of introducing intelligent aids into aircraft cockpits.

By way of caution, it should be noted that whereas covert goal communication imposes little to no additional workload upon humans within the cockpit environment, its use would contribute some degree of misunderstanding of human goals to machine actors. And although a chance of misunderstanding poses only a slight risk in experimental laboratory studies, that slight chance could seriously affect aviation safety in more realistic environments. To overcome this weakness, more reliable methods of overt communication among the human actors should be employed and integrated with the covert goal communication methods. Since the ability of pilots to explain their goals overtly or explicitly to intelligent systems is highly constrained by both time and the limited expressiveness of a non-textual interface, reliance upon such systems leads to increased monitoring workload as certain amounts of inaccurate and improper information are derived from the system. Therefore, it is necessary to develop integrated methods of both overt and covert goal communication, based upon consideration of the tradeoffs between advantages and disadvantages from the selection of each of the two modes.

Most current methodologies for pilot communication do not encompass integrated overt and covert goal communication, often contributing to slow and inappropriate responses to events such as system failures. The principal goal of this research is to present an integrated method for overt and covert (explicit and implicit) goal communication, which can be embedded within a cockpit task management aid for the improvement of CTM performance.

3. RESEARCH OBJECTIVES

The objectives of the present experimental investigation include the following:

1) Develop and present a method to recognize pilot goals in the management of cockpit tasks that is based upon the integration of implicit (covert) as well as explicit (overt) modes of communication;
2) Develop a methodology for accurate goal communication as an embedded routine within an enhanced CTM aid (specifically, the Agenda Manager; see Chapter 4), which will provide Agenda Manager-like aids with an effective tool to resolve goal conflicts between human (pilot) actors and machine actors and eventually to improve Task or Agenda Management performance;
3) Evaluate the methodology presented in a real-time flight simulation environment in terms of its accuracy, speed, user satisfaction, workload and pilot performance of flight control.

The method used to achieve the above objectives is described in the following chapter.

4. METHODS

A methodology embedded within an intelligent cockpit-aid system, providing the ability to recognize and respond to flightcrew cockpit task goals, is presented. The method was developed, implemented, and evaluated in a real-time flight simulation laboratory environment.

4.1. Flight Simulator

The flight simulation environment consisted of three basic software modules as follows:

1) A client program including aerodynamic and autoflight models;
2) A server program including a primary flight display (PFD) with attitude display indicator and horizontal situation indicator without flight path information; and
3) A server program of a secondary flight display with various synoptic aircraft subsystems.

For an aerodynamic and autoflight model, a modified version of the NASA-Langley Advanced Civil Transport Simulator (ACTS) was used. The NASA-Langley ACTS was developed to test flight management concepts and provide a test bed for real-time simulation and flight experiments. The program features a simple aerodynamic and engine model of the airplane with enough control logic to replicate the flight profile of a modern jet transport. The airplane can be flown manually using a pseudo-velocity control mode, automatically utilizing a Guidance and Control Panel (GCP) similar to a Mode Control Panel (MCP), or LNAV (Lateral navigation) and VNAV (Vertical navigation) guidance from a Flight Management Computer (FMC). In the modified version of the ACTS developed for the simulation environment used in the present study, data communication routines were added as client programs and flight displays were simplified greatly.

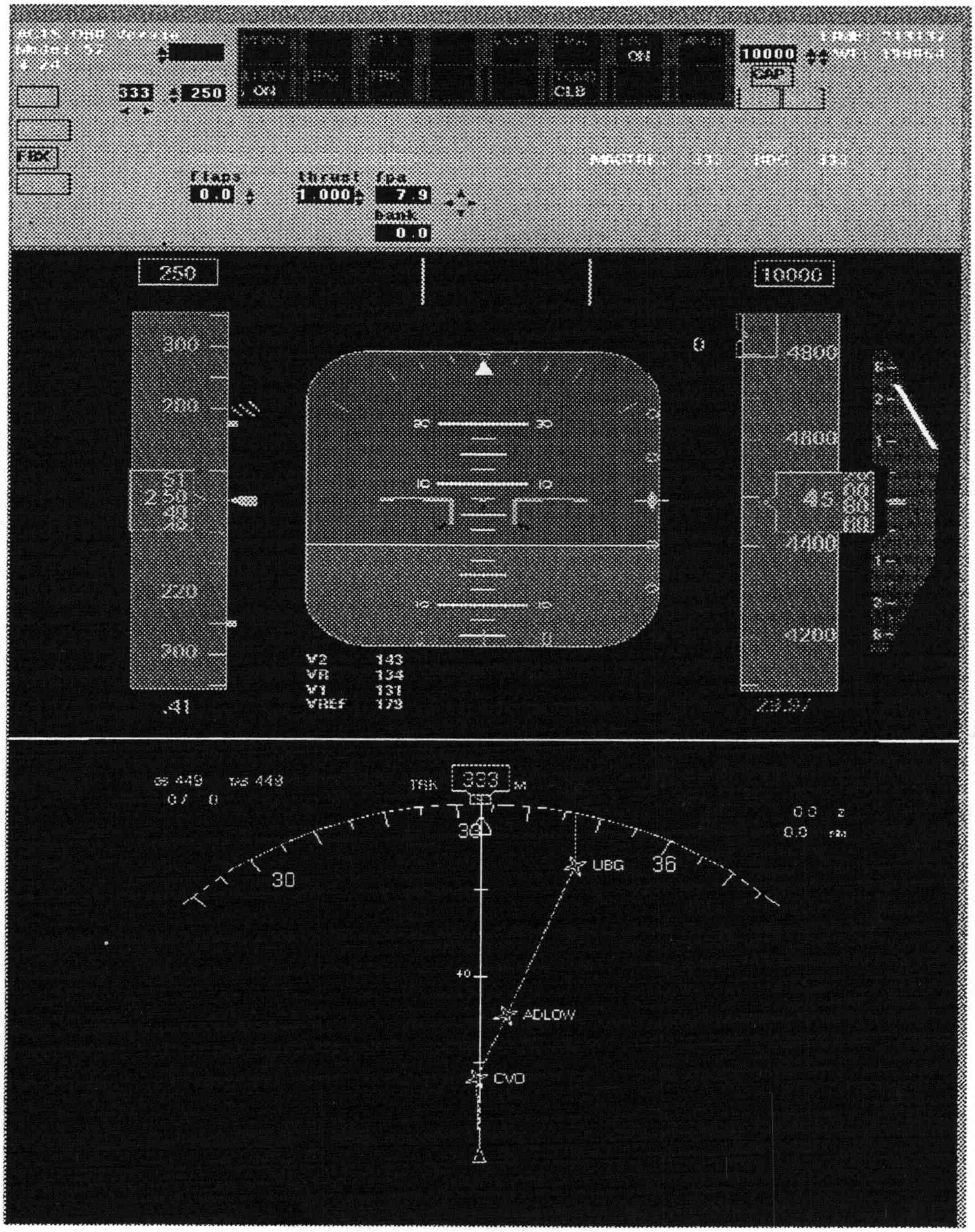

Figure 4.1 Flight Simulator Overview

For primary flight displays, a simplified version of the PFD for a part-task version of the Advanced Concepts Flight Simulator (ACFS) developed by the Man-Vehicle

Systems Research Facility at the NASA Ames Research Center was used due to its similarity to the flight displays of commercial airplanes. As shown in Figure 4.1, this consisted of an Attitude Display Indicator (ADI) and a Horizontal Situation Indicator (HSI). The Flight Path Angle-oriented attitude indicator was used rather than a pitch angle-based indicator; since the aerodynamic model did not provide speed trimming functions and was not a speed-stable model. These NASA-provided software modules were written in C, C++ and FORTRAN.

Secondary flight displays were provided for various aircraft systems, including a fuel system, electric system, hydraulic system, and engine displays. With paged-synoptic capabilities, access was provided to the various event-based aircraft subsystem models and displays shown in Figure 4.2. The program was written in *ParcPlace VisualWorks* 2.51, a visual implementation of Smalltalk, an object-oriented programming language.

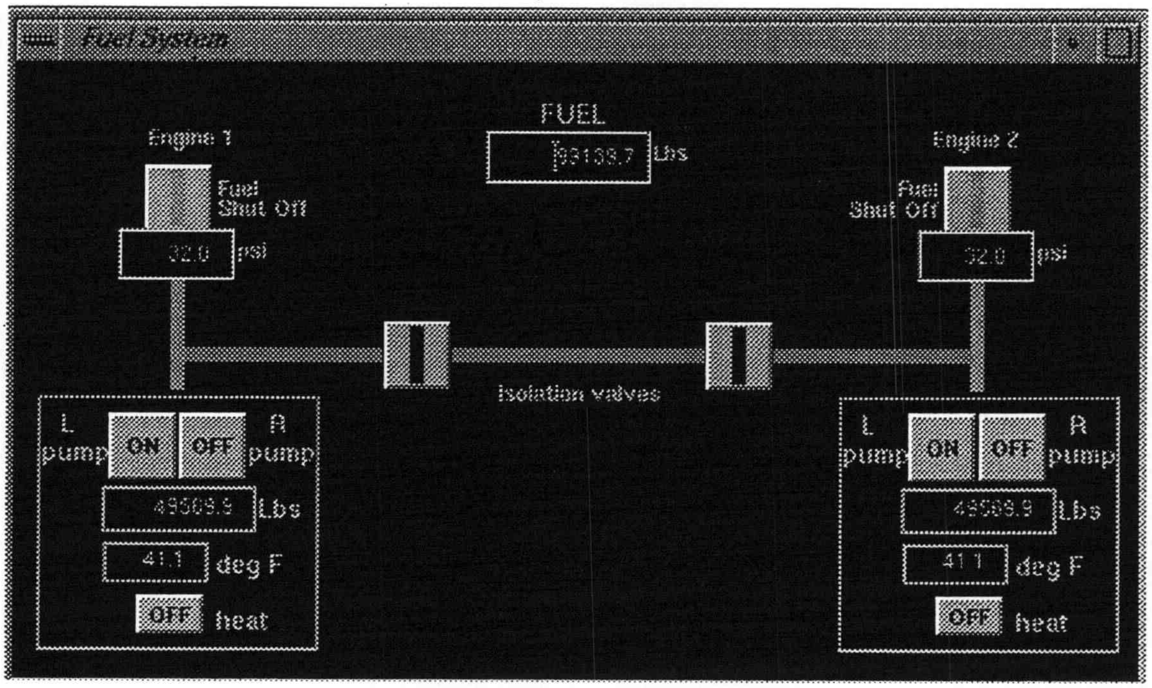

Figure 4.2 System Simulator (Example of Fuel System)

The three software modules were developed based upon a communicating process architecture and a client-server methodology using UNIX sockets. Through the socket connections, this architecture provides flexibility, permitting distributed processing among several workstations to equalize the graphical and computational loads for each component system. Simulator runs were conducted on Silicon Graphics Indigo-2 UNIX-based workstations configured with 200 MHz MIPS 4400 processor, 64 Mbytes of RAM and 1-Mb data/instruction caches. Human subjects interacted with this system via mouse input and/or a BG systems flybox with a three-degrees of freedom joystick with an integral button/lever box providing input to the workstation through an RS-232 serial line.

The integrated flight simulation environment provided a part-task simulator that modeled a two-engine turbojet transport and simulated flightcrew cockpit tasks as well as autoflight and flight-management systems. Thus, the flight simulator provided an appropriate environment for the development and evaluation of the method of cockpit goal communication.

4.2 Agenda Manager, An Intelligent Aiding System

Like human actors, machine actors such as the autopilot or the flight management system, are goal-directed systems which use complex data and knowledge to control aircraft system behaviors. If the object is to accomplish coordinated operations, each of the two types of systems must be knowledgeable about the other's intentions or goals. Based upon these considerations, the CTM study was extended to include agenda management, or the management of associated goals and functions, the actors who perform these functions, and the resources that were used (Funk and Kim, 1995). Managing this agenda is an important process performed by every flightcrew.

A flightdeck agenda consists of a prioritized set of goals to be achieved, a prioritized set of functions to accomplish these goals, a set of actor assignments to functions, and a set of resource allocation to functions. The Agenda Manager (AM, Figure 4.3), a computational aid, was developed to facilitate management of the flightdeck agenda (Funk and Kim, 1995). The AM is an application model in which a goal communication method (GCM) (to be described in the following section) is embedded and in which cockpit agendas are maintained via use of several agents. An agent represents the corresponding entity in the cockpit environment and is implemented as an independent and intelligent software object. The AM consists of the following sets of agents which have both declarative and procedural knowledge:

1) System agents, representing aircraft systems, in which the states and status of the aircraft systems are maintained;
2) Actor agents, representing flightcrew and machine actors, in which current state information of each actor is maintained. The flightcrew agent incorporates intent inferencing as well as explicit goal communication capabilities as an embedded method ;
3) Goal agents representing each state (active, pending or terminated) and the priorities of the goals;
4) Function agents for assessing the functions required to achieve the goals; and
5) An agenda agent, representing the composite flightdeck agenda with which actors' goals and functions are maintained to detect and resolve goal conflicts.

Since this research did not focus on Agenda Manager performance, but rather upon GCM accuracy and its effectiveness in the management of an agenda as an embedded method within AM, a simplified AM was used for the evaluation of the GCM. A fully functional AM, including an interface design, is currently under development (Funk and McCoy, 1996).

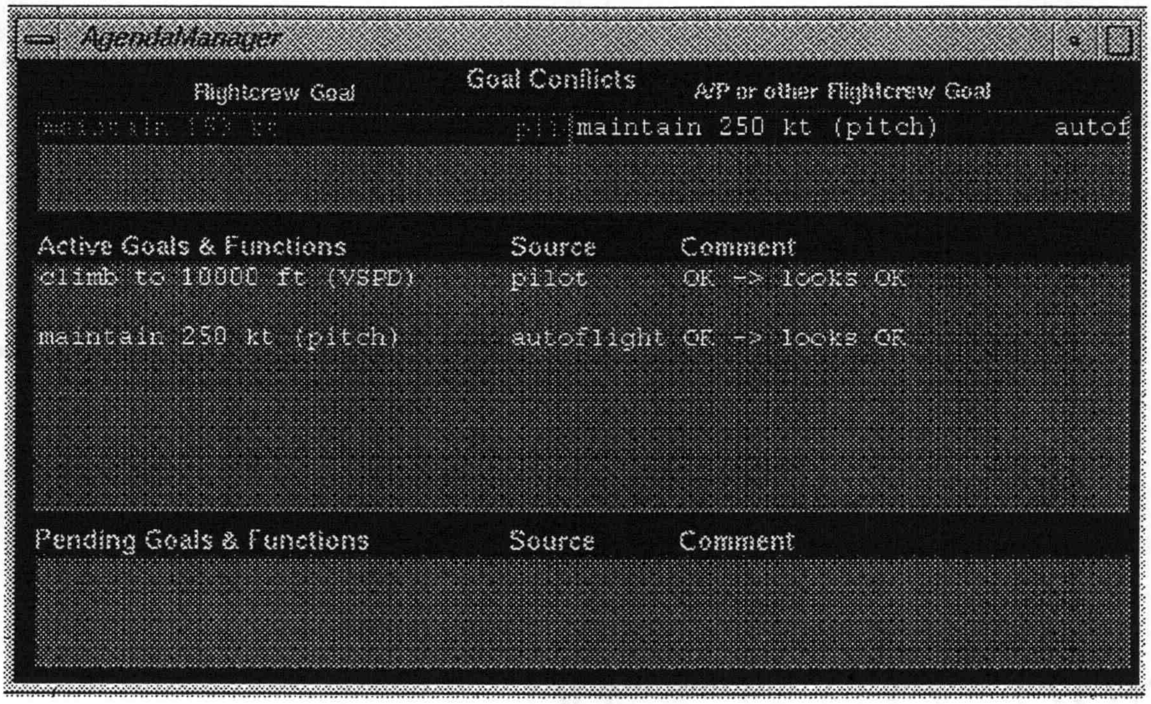

Figure 4.3 Initial version of Agenda Manager Interface

4.3 The Goal Communication Method

The AM serves as an aid to facilitate the management of the flightdeck agenda. To manage this agenda, the AM must have accurate knowledge of pilot goals. For this purpose, the Goal Communication Method (GCM) was installed. The GCM is an embedded method for the recognition, inferencing, updating, and monitoring of pilot goals. The GCM provided a framework for the integration of both implicit and explicit goal declarations into a pilot agenda consisting of the goals, functions, actor assignment to perform functions, and allocation of resources. A method of goal communication between human actors and the AM was installed as a special flightcrew agent. In turn, the flightcrew agent was intended to construct, update, and assess the validity of the pilot agenda, focusing equally upon explicit and implicit methodologies, as specified in the following two sections.

4.3.1 Overt (Explicit) Goal Communication

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

4.3.1.1 Recognition of Pilot Goals

While a pilot is performing flightdeck operations, he or she communicates with an air traffic control (ATC) controller, readily facilitating the detection of pilot goals. Since it is a legal requirement that each pilot read back all ATC clearances, pilot goals can be recognized by the development of a robust method for working with the pilot verbal protocol. For example, if ATC issues the clearance "OSU 037, Eugene ground, climb to 9000," the pilot acknowledges the clearance with a response "Roger, climb to 9000, OSU 037," the ASR system recognizes the pilot voice pattern and signals the flightcrew agent, and then the flightcrew agent parses the signals and declares the goal to the AM.

To interpret the speech patterns of the human actors, the ASR system used was a Verbex VAT31 board (Verbex Voice Systems), which has a 40 MHz Digital Signal Processor (DSP) running under DOS and continuous and speaker-dependent capabilities. With the continuous capability of the Verbex system the subject pilots were able to speak naturally without pauses between words, and with its speaker-dependent capability they were able to easily train the voice recognition system to recognize their specific accents or dialects. The application development processes for voice recognition of the subject pilots using Verbex ASR system were as follow:

1) Creating a grammar definition file (e.g., GCM.grm) containing the speech grammar rules specified in a special notation, called Verbex Standard Notation (VSN) (see Attachment 1). The grammar rules cover those cases that consider pilot habitable vocabulary for a verbal protocol for communicating with air traffic control.

2) Converting the grammar definition file into a recognizer file (e.g., GCM.rec) by compiling the definition file using the Verbex software utility.

3) Transferring the files to the recognizer using the Verbex file transfer software utility. The software utility copies the file into the recognizer's memory.

4) Training and Testing. Subject voice patterns were created throughout the training process. The recognizer was taught the sound of speaker voices from words and phrases in the recognizer file. The information which the recognizer gained from training was stored in a voice file (e.g., CHA.voi).

5) The recognizer was then ready for recognition.

The verbally declared goals were decoded and transmitted to the AM flightcrew agent through the RS232 serial port, as shown in Figure 4.4.

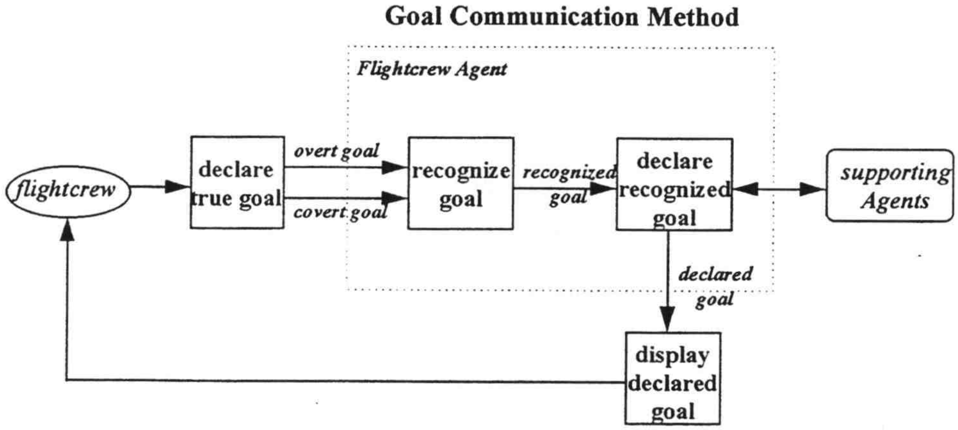

Figure 4.4 Overt goal recognizing process

4.3.1.2 Declaration of Pilot Goals

The encoded form of verbally declared goals was sent through an RS232 serial port to the computer running the AM, in which the goals were parsed, declared, and maintained. As discussed in Section 4.2, the AM was developed using an object-oriented programming environment (specifically, *VisualWorks* 2.51, a visual version of *Smalltalk*), which employs an agent-based architecture consisting of interconnected, independent, intelligent software modules. The flightcrew agent served as an object embedded in the AM for processing the overt goals.

The flightcrew agent encompassed several methods as follows:

a. to connect to the UNIX socket,

b. to parse the decoded overt goal,

c. to declare it as an active goal to the AM in which the goals were maintained, and

d. to update the flightcrew agent interface, as shown in Figure 4.5.

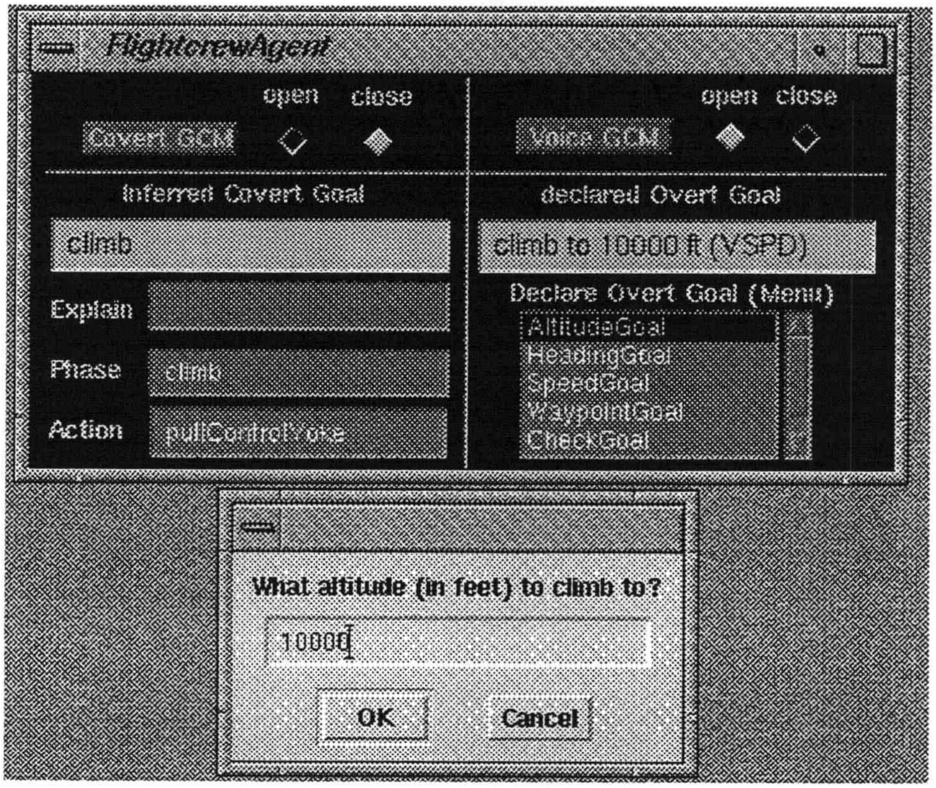

Figure 4.5 Flightcrew Agent Interface

Explicitly declared pilot goals may be considered to be true goals if we assumed that normally a pilot would not say what he/she did not intend to say. Based on this consideration, the declared overt goals provided basic data with which pilot goals might be inferred from the covert GCM described in next section.

4.3.1.3 Efforts to Improve Recognition Accuracy

Accuracy in the recognition of pilot goals is very important. Although accuracy depends to a considerable degree upon current ASR technology, careful human factors engineering of several system design aspects helped to increase recognition accuracy.

4.3.1.3.1 Vocabulary Selection

It has been noted that pilots do not recall correctly all vocabulary items and they are prone to use a *habitable* (i.e., habitual) vocabulary. Thus, it is imperative that the recognizer be able to handle all of the pilot's habitable responses. One way to handle the problem is to ask the pilots to read back exactly what the ATC controller said to them. All ATC verbal protocols are already written in the grammar file. The other way is to allow pilots the use of their habitable phrases and to implement all possible phraseology in a pilot grammar file of the Verbex ASR system which contains rules of speech protocol. For example, the phrases "climb to 9000," "climb and maintain 9000," and "maintain 9000," were all considered and recognized as the same phrases, as specified in the grammar file.

4.3.1.3.2 Training

There are two aspects of training in speaker-dependent system: One is to train the recognizer, a procedure commonly called *enrollment*, the other is to train users. In recognizer enrollment, pilots trained the recognizer in the use of their habitable vocabularies. The template matching to handle the various vocabulary can be enhanced by training users in effective ways of speaking under simulated settings. For example, some pilots preferred to say "niner thousand" for "9000," while some said "nine thousand." Each subject pilot was required to use the phrases in which the recognizer was trained.

In order for the recognizer to detect the exact verbal protocol of a specific goal from the ATC command list phrases, it has been suggested that a pause inserted before and after specific goal phrases is helpful to separate the correct phrase from background noise even though the ASR system used has continuous capability. In training sessions, the subject pilots were trained to emphasize the specific goal phrases in this manner.

4.3.1.3.3 Feedback

Some applications of ASR rely heavily on providing visual and auditory feedback to operators. In addition to the effects of accelerating visual perception, research has shown that visual feedback in ASR is prone to be misleading (Barber et al., 1995). Whenever the recognizer detected a pilot goal, the GCM responded by updating the AM interface and the GCM interface displays. The GCM displayed on the interfaces the exact goal phrase spoken by the pilot. If it recognized the goal, the recognizer and flightcrew agent sounded a beep so that the pilot could confirm the declared goal.

This visual and auditory feedback was important to ensure successful use of overt GCM. This involved devising feedback that was not only easy to confirm, but also was also displayed for a sufficient time so that the pilot did not have to rely on short-term memory. Given the potential problems of feedback monitoring, further research should focus on how to repeat or correct misinterpreted commands.

4.3.1.3.4 Correction of Mis-Recognition

There are two possible outcomes of failed recognition using ASR. One is mis-recognition and the other is incorrect-recognition. When the GCM did not recognize the pilot goal (mis-recognition) through the recognizer, the pilot was not given visual or auditory confirmation. In that case, the pilot simply tried to repeat the goal, speaking in the normal voice used to train the recognizer, until the goals were recognized. Even then, recognition could have been incorrect. To handle the incorrect goals and remove them from the display, pilots had to re-declare the specific goals. If pilots re-declared the goals successfully, the GCM would automatically override the incorrect goals and update the related goals.

4.3.2 Covert (Implicit) Goal Communication Method

While pilot goals were recognized via the overt GCM when communicating with the ATC controller, they were also implicitly inferred from operational and/or other factors, such as pilot actions of moving the control yoke. The method for inferring goals is called covert GCM. As discussed in Chapter 2, covert GCM is important in avoiding the workload associated with overt goal communication, or when it is not possible to communicate goals overtly. Like overt GCM, the covert GCM framework consists of two main parts, one that recognizes goals from an inference mechanism and one that declares the inferred goals to the simplified AM.

4.3.2.1 Recognition of Pilot Goals

To build dynamic representations of current pilot goals, the inference logic for the hypothesized current pilot intentions was based upon the following four components:

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

With knowledge of the above four components, a script was constructed as a data-driven knowledge source. The script consisted of a representation of loosely ordered sets of pilot actions to carry out a particular goal. Given the current state of the above component variables and flight phases, GCM tried to interpret pilot actions based upon script-based reasoning processes. An example 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
```

If the action could be explained by an active script, the corresponding active goal was recognized and declared by the intent inferencer, which represented a process model using a blackboard problem-solving method. The knowledge source in this blackboard framework consisted of a rule-based representation of goals and corresponding scripts for a given domain.

If the actions were not predicted by the active script, then the GCM would attempt to ask the pilot to ignore the covert GCM and declare his or her goal explicitly using overt GCM.

4.3.2.2 Declaration of Inferred Pilot Goals

The flightcrew agent served as one of the classes embedded in the AM for processing explicitly inferred goals. The flightcrew agent declared a goal to be active, passed it to the agent in which the goals were maintained, and then updated the flightcrew agent interface. The covert GCM process is shown in Figure 4.6.

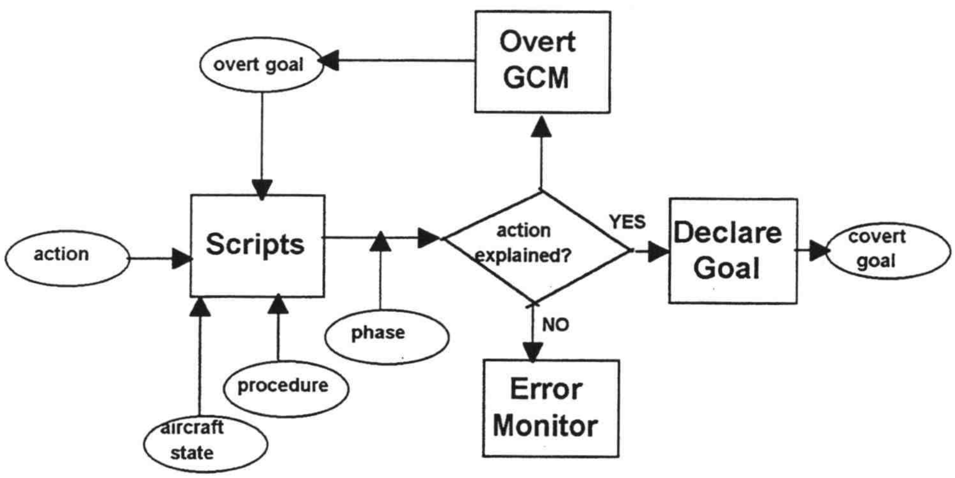

Figure 4.6 Covert GCM process

4.3.2.3 Efforts to Improve Recognition Accuracy

Obviously, the accuracy of the covert GCM was dependent upon the content of the scripts, each of which contained domain knowledge. Robust domain knowledge should be obtained from flight expertise, including the flight operations manual. Thus, accuracy could be increased in accordance with how well the knowledge base covered the expert knowledge. To increase accuracy, covert GCM was reviewed by a domain expert, specifically, a former Boeing flight test pilot. Revisions to the GCM were made based upon his recommendations.

For incorrectly inferred goals, the pilot could terminate the covert GCM process or ignore it and declare goals explicitly using overt GCM.

4.4 Evaluation of the GCM

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

4.4.1 Subjects and Scenarios

The GCM was evaluated by 10 licensed general aviation pilots. Although most did not have commercial licenses and were not familiar with the ADI and HSI displays, all had some instrument flying knowledge and experience in controlling and monitoring aircraft altitude, speed, and heading. All of the subjects also had experience in ATC communication. For the experiments, two subjects were used for a readiness test of GCM implementations to be used and the remaining eight subjects were used for data collection. Of the latter, each was asked to perform two simulated flight task scenarios based on a Eugene, Oregon to Portland, Oregon flight (see appendix). As shown in Table 4.1, the scenarios contained a number of goals to be declared and performed by each subject. Overt goals were to be declared by the pilot (as observed by ATC) and covert goals were to be inferred by GCM. Differences in phraseology between overt and covert goals reflected different pilot's habitually used vocabularies.

Table 4.1 Declared goals, Eugene-Portland scenario.

Phase	*Overt Goal*	*Covert Goal*
before takeoff	*maintain 9000*	*climb to 9000*
	runway 21	*maintain heading 210*
		attain Vr
takeoff		*maintain speed v2+15*
		maintain climb speed 210
climb	*turn right heading 290*	*maintain heading 290*
	turn right heading 330	*maintain heading 330*
	direct Corvallis	
	maintain speed 240.	*maintain 9000*
	climb and maintain 14000	*climb to 14000*
		maintain 14000
cruise	*maintain current heading 352*	
	maintain speed 280	*maintain cruise speed 280*
descend	*reduce speed to 240*	
	descend to 10000	*maintain speed 240*
	direct UBG	*maintain 10000*
approach	*turn left heading 334*	*maintain heading 334*
	turn right heading 360	*maintain heading 360*
	descend to 7000	*maintain 7000*
	reduce speed to 210	*maintain speed 210*
	turn right heading 010	*maintain heading 010*
	descend and maintain 4000	*maintain 4000*
	turn right heading 070	*maintain heading 070*
	maintain 3000	*maintain 3000*

As shown in Table 4.2, all of the above goals can be categorized as either altitude, speed, or heading goals.

Table 4.2 Declared goals classification, Eugene-Portland scenario.

Category	*Overt Goal*	*Covert Goal*
Altitude Goal	*maintain 9000*	*climb to 9000*
	climb and maintain 14000	*maintain 9000*
	descend to 10000	*climb to 14000*
	descend to 7000	*maintain 14000*
	descend and maintain 4000	*maintain 10000*
	maintain 3000	*maintain 7000*
		maintain 4000
		maintain 3000
Speed Goal	*maintain speed 240.*	*attain Vr*
	maintain speed 280	*maintain speed v2+15*
	reduce speed to 240	*maintain climb speed 210*
	reduce speed to 210	*maintain cruise speed 280*
		maintain speed 240
		maintain speed 210
Heading Goal	*runway 21*	*maintain heading 210*
	turn right heading 290	*maintain heading 290*
	turn right heading 330	*maintain heading 330*
	direct Corvallis	*maintain heading 334*
	maintain current heading 352	*maintain heading 360*
	direct UBG	*maintain heading 010*
	maintain heading 360	*maintain heading 070*
	turn left heading 334	
	turn right heading 360	
	turn right heading 010	
	turn right heading 070	

4.4.2 Procedures

To measure GCM effectiveness in terms of accuracy and workload, subjects were required to fly the Eugene-to-Portland scenario manually involving a number of goals (Tables 4.1 and 4.2) to be declared by them and containing no autoflight goals. Based upon use of the same scenario with the same flight simulation conditions, one experiment was performed running with the GCM and a second running without the GCM. To eliminate a learning effect between the two experiments, the order of experiments was balanced. That is, half of the subjects performed the GCM experiment first whereas the other half performed first without use of the GCM. The GCM was installed with a simplified AM interface (Figure 4.7) rather than a fully functional AM. The simplified AM interface did not provide subject pilots with such agenda management information as function assessment, but simply displayed the declared goals. The intent underlying this approach was to eliminate the possible compounding effects of agenda management in measuring pilot performance.

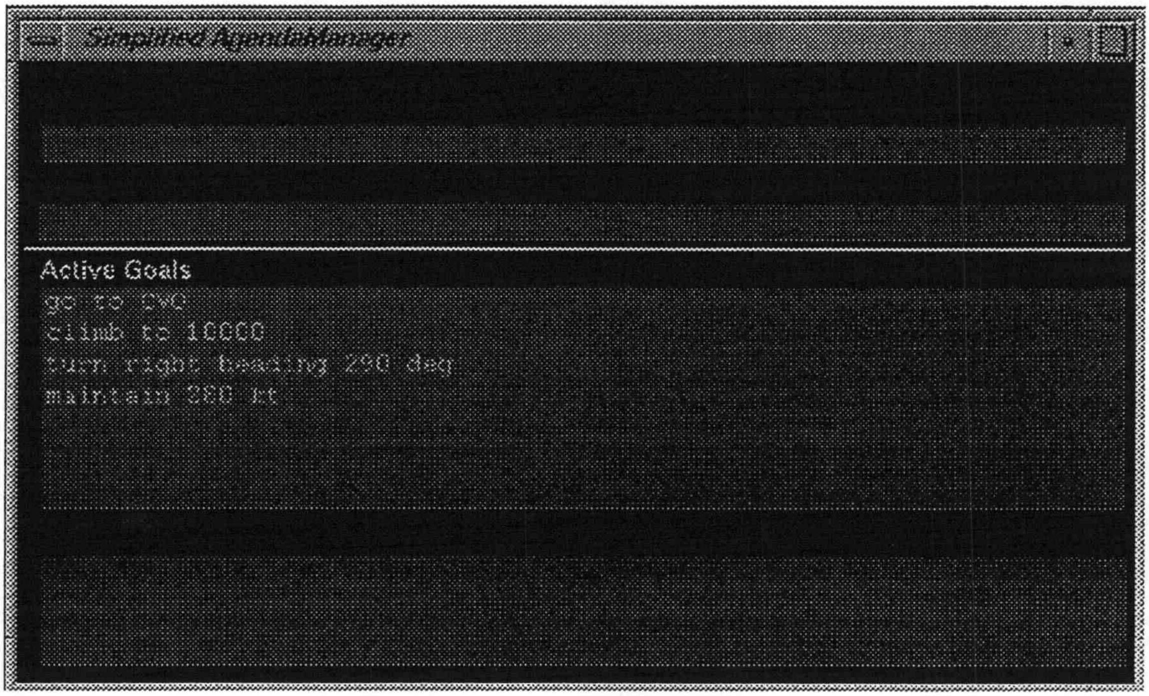

Figure 4.7 Simplified Agenda Manager interface for GCM evaluation.

The AM assumed that ATC commands for the scenario were the true goals that the subjects should attain. The experimenter played the role of ATC controller. Based on ATC commands, the subjects declared their goals verbally and/or with related flight control actions. The GCM captured the raw form of the declared goals, parsed them into meaningful phrases, and then interpreted and displayed the goals.

The subject pilots called out their goals explicitly using a headset microphone. Speech patterns were collected from the subjects concurrently, as they verbalized their intentions, actions, and problem-solving activities while operating the flight simulator system models. Prior to the simulated flight, the subjects were required to train the recognizer to adapt to their voices in two 20-minute training sessions. In the training sessions, the subjects twice read back 187 simple phrases displayed on the monitor (e.g., "turn left heading 290"). While they were flying, subjects were supposed to read back ATC commands immediately after they were heard. If they failed to declare their goals verbally, they were asked to repeat their goals until the overt GCM recognized them. The successfully declared goals were displayed on the interface displays. The subjects also removed their goals verbally whenever this was required. The context and grammar files for verbal protocols are included in the appendix. To measure pilot performance in the experiments without GCM use, the experimenter's voice as the ATC controller was used for declaring goals.

The subject goals were also declared and recognized via covert GCM, which employed an intent-inferencing mechanism based on aircraft states, subject control actions, and verbally-declared active goal as described above. Whenever the subjects took actions using thrust levers or control buttons and levers, the GCM inferred, interpreted and displayed the goals. The covert GCM interface module automatically displayed information about subject actions, the flight phase, and the inferred goals. A statement of intentions for each action was compared to the interpretations of pilot agents employing the GCM (i.e., scripts). If there were conflicts between subject and GCM interpretations, the covert GCM indicated "PilotActionNot Understood" with a warning sound. Whenever the subjects heard a warning sound due to actions that could not be understood, they were asked to remove the conflict by taking a corrective action. If the GCM understood the action or corrective action, it was considered that the GCM recognized the subject pilot's goals correctly, and then it would display a correctly inferred goal. If the covert GCM failed to recognize the goal correctly, subjects were required to declare the goal verbally using overt GCM

When subjects reached a designated point in a scenario, the experimenter stopped the simulator and asked questions to collect such data as perceived workload and whether the GCM had recognized the goals accurately. To measure subject workload when using the GCM, the NASA-TLX multi-dimensional subjective measure was used (Figure 4.8), as developed at the NASA Ames Research Center (Hart and Staveland, 1988). More detailed information is provided in the appendix. The experimenter asked the subjects to rate their workload using a 10-point scale (from 0 = easy to 10 = hard) in a pop-up workload measurement window. Weighted averages for ratings on six subscales were automatically calculated and recorded in a log file. The experimenter, seated beside the subject pilot and issuing ATC commands, was responsible for collecting subject workloads and determining whether the GCM recognized goals correctly. To ensure accurate and objective experimental analysis, the entire flight simulation was videotaped.

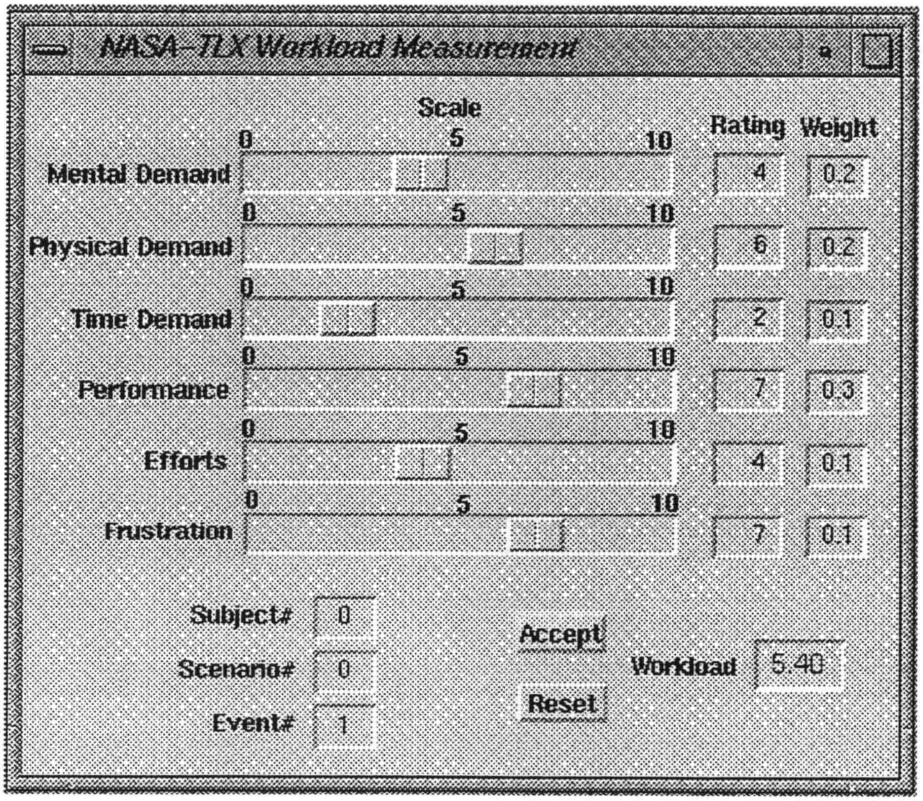

Figure 4.8 NASA-TLX Workload Measurement Interface Display

4.4.3 Dependent measures.

As a specific dependent measure to test GCM accuracy, the number or percent of successful goal recognitions was measured. For use of overt GCM, one criterion for the successful recognition of goals was straightforward. If the subjects called out their goal and the GCM displayed it as spoken, goal recognition was considered to be successful. For the covert GCM accuracy measure, while the simulator was running the subjects were asked questions about the goals they tried to achieve. Their intended goals were compared with those inferred by the GCM. If there was no conflict between the two, recognition was counted as successful.

Subjective workload was measured as an overall workload score based on weighted average ratings for six subscales: mental demands, physical demands, time demands, performance, efforts and frustration. A 10-point workload scale (from 0 = easy to 10 = hard) was used for each subscale rating. The subscale ratings were weighted according to their subjective importance to the specific task situation.

Pilot performance was measured using functional assessment information for each declared goal. In the AM, pilot actions to achieve a declared goal were maintained by a function agent assessing whether performance was satisfactory, marginal or unsatisfactory. Since speed, altitude and heading goals are always maintained during a flight simulation, the portion of the time the function was satisfactory was used as a pilot performance measure. The functional status data of a declared goal was collected whenever the goal was updated (approximately 30 Hz).

4.4.4 Statistical Analysis.

For GCM accuracy, confidence interval estimation was used as a statistical analysis tool. Confidence intervals were presented as mean values of a percentage of the number of successful recognitions for declared speed, altitude and heading goals.

For workload measurement, the null hypothesis was that there would be no correlation between any predicted workload running either with the GCM or without the GCM. In other words, the issue of concern was whether extra workload could be expected when the GCM was used. Parametric statistical analysis, a paired *t*-test, was then conducted for paired findings and compared to experimental data to determine if use of the GCM had a significant effect on the level of workload.

For pilot performance measurement, the null hypothesis was that there would be no correlation between predicted pilot flight control performance when running either with or without the GCM. A paired *t*-test was then conducted for the paired findings and compared to experimental data to determine if use of the GCM had a significant effect on pilot flight control performance.

5. RESULTS

This chapter presents the results of the experiment and an analysis of the results. The experimental results provided in the first section are based upon subject performance data obtained from the data collection sessions. The second section provides a description of how the aggregated results were analyzed with respect to the evaluation of GCM effectiveness.

5.1 Experimental Results

Following the procedures discussed in Chapter 4, each of the eight subjects performed two experiments with the identical flight scenario: four subjects performed the with-GCM-experiment first and the remaining four subjects performed the without-GCM-experiment first to eliminate any possible learning effect. The number of failed recognitions was recorded by the experimenter during flight simulation and confirmed by videotape analysis following flight simulation. The pilot performance data, based on functional assessment of each declared goal, were automatically recorded on log files while the simulation was running. The workload data, based on weighted average ratings for six subscales, were recorded by the subjects when the simulation was temporarily stopped. The summary of experimental results is shown in Tables 5.1-5.4.

Table 5.1. Experimental results of eight subjects for the remaining number of failed recognitions after the **first** declaration of goal.

Selected Method	Declared Goals	Subjects 1	2	3*	4*	5	6*	7	8*	tot
overt	*takeoff & climb (7)*	*2*	*1*	*2*	*0*	*1*	*1*	*1*	*0*	*6*
	cruise & descend (5)	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
	approach (8)	*1*	*1*	*1*	*0*	*1*	*0*	*1*	*1*	*6*
	accuracy (%)	***85***	***90***	***85***	***100***	***90***	***95***	***85***	***95***	***91***
covert	*takeoff & climb (10)*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*0*	*2*
	cruise & descend (3)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	approach (8)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	accuracy (%)	***100***	***96***	***100***	***100***	***96***	***100***	***100***	***100***	***99***
integrated	*# of failed recognition*	*3*	*3*	*2*	*0*	*3*	*1*	*2*	*1*	*15*
	# of total declared goal	*41*	*41*	*41*	*41*	*41*	*41*	*41*	*41*	*328*
	accuracy (%)	***92***	***92***	***95***	***100***	***92***	***98***	***95***	***98***	***95***

Note: the number within parentheses is the total number of declared goals of that type.
* indicates the subjects who performed with-GCM-experiment first.

Table 5.2. Experimental results of eight subjects for the number of failed recognitions after the **second** declaration of the first failed goal.

Selected Method	Declared Goals	Subjects 1	2	3*	4*	5	6*	7	8*	tot
overt	*takeoff & climb (7)*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*2*
	cruise & descend (5)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	approach (8)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	accuracy (%)	***96***	***100***	***96***	***100***	***100***	***100***	***100***	***100***	***99***
covert	*takeoff & climb (10)*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	cruise & descend (3)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	approach (8)	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
	accuracy (%)	***100***	***100***	***100***	***100***	***100***	***100***	***100***	***100***	***100***
integrated	*# of failed recognition*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*2*
	# of total declared goal	*41*	*41*	*41*	*41*	*41*	*41*	*41*	*41*	*328*
	accuracy (%)	***98***	***100***	***98***	***100***	***100***	***100***	***100***	***100***	***100***

Note: the number within parentheses is the total number of declared goals of that type.
* indicates the subjects who performed with-GCM-experiment first.

Table 5.3 Experimental results for weighted workload for eight subjects.

Condition	*Legs*	*1*	*2*	*3**	*4**	*5*	*6*	*7**	*8**	*avg*
With GCM	*takeoff & climb*	3.8	6.0	4.1	4.2	2.4	2.3	3.5	4.2	3.8
	cruise & descend	0.5	2.6	1.2	0.2	1.2	1.5	1.5	2.0	1.3
	descent & approach	5.5	7.6	4.2	4.1	1.8	4.9	6.1	4.5	4.8
Without GCM	*takeoff & climb*	4.9	3.3	2.2	2.1	2.7	1.2	3.8	3.5	3.0
	cruise & descend	1.3	2.6	0.8	1.0	0.2	0.7	1.2	1.8	1.2
	descent & approach	7.1	7.6	4.8	1.7	2.9	1.2	5.9	4.3	4.4

0 = low, 10 = high workload

* indicates the subjects who performed with-GCM-experiment first.

Table 5.4 Experimental results for pilot flight control performance, percent of total simulation time in managing goal with ***satisfactory performance*** for eight subjects.

Condition	*Declared goals*	*1*	*2*	*3**	*4**	*5*	*6**	*7*	*8**	*avg*
With GCM	*Speed goals*	70%	76%	72%	60%	66%	70%	65%	68%	68%
	Altitude goals	37%	51%	45%	39%	40%	40%	45%	44%	43%
	Heading goals	48%	69%	56%	50%	41%	42%	55%	49%	51%
Without GCM	*Speed goals*	62%	58%	70%	71%	46%	69%	60%	72%	64%
	Altitude goals	35%	48%	52%	41%	37%	39%	42%	48%	43%
	Heading goals	40%	52%	60%	49%	30%	47%	50%	54%	48%

* indicates subject who performed with-GCM-experiment first.

5.2 Analysis of Results

5.2.1 Recognition Accuracy

One of the objectives of this research was to develop an accurate goal communication method. GCM accuracy was measured statistically using confidence-interval estimation to determine accuracy as follows.

Assume that X is a normally distributed random variable with unknown GCM recognition accuracy mean and variance μ and σ^2, respectively. A random sample of small size 8, X_1, X_2, ..., X_8 is available, and $\bar{X}$ and S^2 denote the sample mean and sample variance of recognition accuracy of GCM, respectively. With the assumption of normality for small sized X, the sampling distribution of the statistic

$$t = (\bar{X} - \mu)/(S/\sqrt{n})$$

is the t distribution with n-1 degrees of freedom. Letting $t_{\alpha,n-1}$ 100(1- α) percent lower confidence interval on t so that

$$P\{\, t \le t_{\alpha,n-1} \,\} = 1 - \alpha$$

or $$P\{\, (\bar{X} - \mu)/(S/\sqrt{n}) \le t_{\alpha,n-1} \,\} = 1 - \alpha .$$

Rearranging the equation yields

$$P\,\{\, \bar{X} - t_{\alpha,n}\, S/\sqrt{n} \le \mu \,\} = 1 - \alpha .$$

Hence, a 95 percent lower one-sided confidence interval on the recognition accuracy of overt GCM was computed as follows:

$$\bar{X} - t_{.05,7}\, S/\sqrt{8} \le \mu$$

or $87\% \le \mu$, where average recognition accuracy is 91%.

And also, a 90 percent lower one-sided confidence interval on the recognition accuracy of overt GCM yielded

$$\bar{X} - t_{.01,7}\, S/\sqrt{8} \le \mu$$

or $89\% \le \mu$, where average recognition accuracy is 91%.

Similarly, a 95 percent lower one-sided confidence interval on the recognition accuracy of the integrated method of covert and overt GCM was computed as follows:

$$93\% \leq \mu, \text{ where average recognition accuracy is } 95\%.$$

And also, a 90 percent lower one-sided confidence interval on the recognition accuracy of the integrated method of covert and overt GCM yielded

$$94\% \leq \mu, \text{ where average recognition accuracy is } 95\%.$$

Therefore, it may be stated that the GCM accuracy for the recognition of overt goals was at least 87% with $\alpha = 0.05$ (i.e., probability of type I error) while the average recognition accuracy was 91%. In other words, 87% of the explicitly declared goal were successfully recognized by the first GCM process with 95% level of confidence. Although it is technically difficult to obtain 100% accuracy with current ASR technologies, the experimental results showed that the second trials of overtly failed recognized goals achieved close to 100% accuracy. Thus, if we accept the cost of the second trial compared to the benefit of employing GCM, the GCM being considered can be used accurately as an embedded method to communicate pilots goals explicitly within the cockpit task simulation environment. Additional details will be discussed in the next chapter.

Since the covert GCM produced almost 100% accuracy in inferring pilot goals, the accuracy of the integrated method of overt and covert GCM was increased to 94%. The high accuracy of the covert GCM results from the fact that the goals evaluated for the present study were based on limited actions and simple scripts and rules. In this relatively simple flight environment, compared to a complex environment which required complex rules and facts to infer pilot goals, inferencing errors were less likely to occur. Althoughlimited by available resources, the real flight environment developed for these experiments provided a sufficient test to demonstrate the benefits of using integrated methods in a complex, dynamic aircraft system. Additional details will be discussed in the next chapter.

5.2.2 Comparison of workload

The objective of measuring workload was to know if any additional workload was imposed on subjects using GCM. As shown in Table 5.5, as based upon a set of eight paired observations, it was assumed that the differences were normally and independently distributed random variables with mean μ_D and variance σ_D^2.

Table 5.5 Workload comparison chart.

	speed goal			*altitude goal*			*heading goal*		
	w/ GCM	*w/o GCM*	*difference*	*w/ GCM*	*w/o GCM*	*difference*	*w/ GCM*	*w/o GCM*	*difference*
mean	3.8	3.0	0.9	1.3	1.2	0.1	4.8	4.4	0.4
variance	1.36	1.34	1.80	0.59	0.54	0.44	2.85	5.67	3.20
t_0			1.791			0.588			0.633
$t_{.05,7}$			1.895			1.895			1.895

To test

$$H_0 : \mu_D = 0$$

$$H_1 : \mu_D > 0$$

the test statistic used was

$$t_0 = \bar{D} / (S_D / \sqrt{n})$$

where $\bar{D}$ and S_D were, respectively, the sample mean and variance of the differences.

That is, the null hypothesis was that there would be no difference between predicted workload running either with or without the GCM. In other words, the null hypothesis was that there would be no additional workload when subjects used GCM.

From the results shown in Table 5.5, the null hypothesis, H_0, cannot be rejected. Therefore, it may be safely concluded that no extra workload will be required during flight simulations when GCM is used.

5.2.3 Comparison of pilot flight control performance

The objective of measuring pilot performance in controlling flight was to know whether the use of GCM had an effect on pilot performance in controlling flight. Table 5.6 compares the data collected with and without GCM as a percentage of satisfactory performance. It was assumed that the differences were normally and independently distributed random variables with mean μ_D and variance σ_D^2.

Table 5.6 Flight control performance comparison chart

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

To test

$$H_0 : \mu_D = 0$$

$$H_1 : \mu_D \neq 0$$

the test statistic used was

$$t_0 = \bar{D} / (S_D / \sqrt{n})$$

where $\bar{D}$ and S_D were, respectively, the sample mean and variance of the differences.

That is, the null hypothesis that there would be no difference between predicted performances in performing flight control either with or without the GCM could not be rejected. Therefore, it may be concluded within reason that the use of GCM did not significantly affect pilot performances during flight simulation.

6. DISCUSSION

Overall, the laboratory experiments conducted for the present study demonstrated the ability of the GCM to communicate overt and covert goals. Specifically, the overt and covert integrated method achieved 95% accuracy while the overt GCM alone obtained 91% accuracy. It was also indicated that the GCM neither imposed extra workload on the subjects, nor affected subjects' flight control performance. However, this is not to say that the GCM would not face potential limitations when applied to real flight systems. The problems and limitations of the GCM used for this study with respect to use in future flight simulation studies or in real systems may be solved by the inclusion within the GCM of certain recently developed technological advances.

6.1 Extension To ASR Technology Update

Over the past two decades advances in ASR technology have contributed to a technology that may be applied to aviation domains exhibiting mentally, physically and psychologically stressful environments. The success of declaring pilot goals overtly is thus dependent upon current ASR technology, although at the same time, the accuracy level of ASR technology could be increased further by the implementation of the means of recognition accuracy developed for the present study. As seen from the experimental results, approximately 9% of the first goal declaration trials failed recognition tests, indicating that further advances in the ASR technology will be critical to achieve increased recognition accuracy for declared pilot goals.

In addition to advances in ASR technology, further efforts will be required to produce recognition algorithms within current levels of ASR technology that are sufficiently robust to deal with potential flightdeck operation communication problems, such as improper use of ATC command verbal protocols. Solution of the problems that occur in laboratory simulations can be approached by the use of vocabulary selection based upon greater variance and covering all possible *habitable* (i.e., habitual) pilot vocabularies; the design of training procedures with broader applicability; and the employment of human factors principles in the design of feedback displays.

Although the GCM and other systems using ASR (e.g., CASSY, Onken, 1995; Gerlach et al., 1995) were reasonably successful in flight simulation environments, the degree of recognition accuracy achieved may not be sufficient to merit application to systems or methods used in real environments (i.e., the CASSY speech recognition accuracy was 88%). Nevertheless, several investigations have successfully used the ASR system for the recognition of overtly declared pilots goals in real cockpit environments, leading to the overall conclusion that most overt goal recognition errors could be removed by repeating declarations of unrecognized goals or by the application of currently updated ASR technologies (Williamson, 1996; Onken, 1995). In fact, the experimental results from the present study demonstrated that the second trials for failed recognized goal achieved close to 100% accuracy. At the same time the ASR system used, the Verbex VAT31, was so outdated that if more highly developed technologies (i.e., natural language understanding capability) were employed, recognition accuracy would be increased to an even greater extent.

Thus, if we accept the costs of second trials or of the inclusion of advanced technologies, compared to the benefit of employing the GCM, the GCM can be accurately used within the cockpit task simulation environment as an embedded method to communicate pilots goals explicitly.

6.2 Extension To a Robust Intent Inference Model

It has been recognized that in such a complex, dynamic domain as aviation, use of model-based inferencers have often contributed to some degree of misunderstanding of human goals due to their inability to accommodate all possible unexpected actions by human pilots (Rouse, 1988; Rubin et al., 1988; Callantine & Mitchell, 1994). Although the chance of misunderstanding poses only a slight risk in experimental laboratory studies, that slight chance may have serious effects upon aviation safety in more realistic environments. To overcome this limitation, the present study sought to demonstrate the ability to use overt GCM means to recognize incorrectly inferred goals.

From the experimental results, the GCM covert goal-recognizing ability compared favorably to human interpretations of the same actions. But this almost perfect match (i.e., 99% accuracy) resulted in the technical problem that the experiment provided too few opportunities to demonstrate the GCM ability to use overt goals when pilot goals were incorrectly inferred. Regardless of the supplemental experiment to demonstrate the GCM ability to handle incorrect covert goals from use of overt goals, the GCM demonstrated inferred goal overriding capability simply by replacing an incorrectly inferred goal with next overt goal on the scenario. The accuracy of the covert GCM thus seemingly resulted from the fact that the inferred goals evaluated for the present research, due to limitations upon available resources, were based on limited actions, simple scripts and rules, and simple scenarios. Therefore, inferencing errors were less likely to occur within the relatively simple flight environments used when compared to complex environments based upon use of numerous rules and facts to infer pilot goals.

To provide an enriched environment for the study of pilot intent inferencing in GCM which is applicable to more complex aircraft systems, it will be necessary to update the current scripts and rules to obtain greater degrees of confidence in experimental findings, as well as to employ a robust Intent Inferencing Model such as the OFM (Operator Function Model) or the HM (Hazard Monitor, Skidmore et. al., 1996).

6.3 Future Research

Given that GCM goal recognition ability is based on a current ASR system, and that an intent inferencing mechanism using simple scripts and rules has been validated, the next step is to implement the GCM within a fully functional agenda management aid, the Agenda Manager, which can provide functional assessment routines for all recognized and declared goals. This will allow the GCM to run on the Agenda Manager as an embedded method for the communication of pilot goals and for the evaluation of agenda management activities involving machine actor assignments (i.e., the autopilot). Hence, this step will provide a simulation environment for the evaluation of goal conflicts between human pilots and machine actors.

This experiment was performed in a laboratory simulation environment which provided different flight simulation conditions from the real environments which produce such stressful situations as high noise levels and high acceleration forces. These stressful situations impair the use of working memory and will eventually reduce the recognition accuracy of the overt GCM. Although it has been suggested that this possible weakness could be confronted by comprehensive user training (Baber, 1991; Williamson, 1996), future research should be focused upon undertakings that specifically address the effects of stressful situations on the overt GCM.

6.4 Conclusion

Based upon the assumption that if an intelligent cockpit-management aid encompassing the ability to accurately recognize pilot goals while monitoring the functions performed to achieve those goals could be developed, this cockpit task management (CTM) aid could be used to improve agenda management performance, bringing goal conflicts, unsatisfactory function performance, and other problems to the attention of the aircraft pilot (Funk & Kim, 1995; Funk & McCoy, 1996). The experimental development of the GCM, a method for the recognition, declaration, and evaluation of pilot goals embedded within a simplified agenda management aid, directed at the evaluation of a performance management agenda based upon a set of goals, functions, actor assignments and resource allocations during aircraft flightdeck operations, were thus among the objectives of the present study

Insofar as it was demonstrated that the GCM developed for the present study has the capacity to recognize pilot goals with a certain degree of accuracy as well as to accommodate incorrectly declared goals, its usability in a simple yet realistic flight simulation environment has been clearly established by laboratory experimentation indicating that the GCM would have no effect upon workload and pilot control performance. Therefore, the findings from the present study have indicated that the GCM may be implemented within the Agenda Manager as a method to ascertain flightcrew goals, thus facilitating the agenda management tasks in cockpit operations.

Despite this successful demonstration of GCM usability, the issue of the optimal level of pilot goal recognition accuracy required to cope with unexpected events in real flight environments remains a continuing research concern. For example, though a machine actor can be developed that has the capacity to understand human intentions with reasonable levels of recognition accuracy (e.g., 99.9% accuracy), the slightest errors could exercise serious effects upon aviation safety in more realistic environments or real-time flight activities. To further minimize the possible effects of human error, various research efforts, such as those described in the previous section, should be undertaken in this field.

Findings from the present study have contributed to research efforts reinforcing ultimate human responsibility for the control of cockpit operations. Thus, in the event of unsatisfactory levels of GCM recognition accuracy, human means can be used to assume control from mechanized systems, changing the human role from that of passive monitor to active controller. This is in general agreement with the current trends in human factors engineering directed at the development of human-centered uses of automation (Wickins, 1992; Billings, 1996; Riley, 1996). In this sense, the GCM is based upon a human-centered conceptualization. That is, when GCM performance was less than satisfactory, human actors were permitted to terminate the GCM process and to actively assume control of the machine actors. The GCM, as implemented for the Agenda Manager as well as human-centered pilot goal communicating methods, will provide a rich environment for the study of human-machine interactions in the supervisory control of complex dynamic systems.

REFERENCES

Aerospace (1987) Coming: Cockpits that hear, *AEROSPACE* design News, April 6, 1987, pp. 72-76.

Aviation Week & Space Technology (1994) "Autopilot Go-Around Key To CAL Crash" pp. 31-32, May 9, 1994.

Aviation Week & Space Technology (1996) "Pilots, A300 Systems in Nagoya Crash" pp. 36-37, July 29, 1996.

Baber, C., Usher, D.M., Stammers, R.B. and Taylor, R.G. (1992). Feedback requirements for automatic speech recognition in the process control room. *International Journal of Man-Machine Studies*, 37, 703-719.

Baber, C. and Noyes, J. (1996) Automatic Speech Recognition in Adverse Environments. *The Journal of the Human Factors and Ergonomics Society*, 1996, 38(1), 142-155.

Billings, C.E. (1991) Human -Centered aircraft automation: a concept and guidelines *(NASA Contractor Report 103885*). Moffett Field, CA: NASA Ames Research Center.

Billings, C.E. (1996) Human -Centered aviation automation: Principles and guidelines *(NASA Contractor Report 11038*). Moffett Field, CA: NASA Ames Research Center.

Boeing Commercial Airplane Group (1994) Statistical Summary of Commercial Jet Aircraft Accidents: Worldwide Operations, 1959-1993. Seattle: Airplane Safety Engineering (B-210B).

Callantine, T.J., Mitchell, C.M. (1994). A methodology for Understanding How operators Select and Use Modes of Automation to Control Complex Dynamic Systems. *Proceedings of the 1994 International Conference on Systems, Man, and Cybernetics*, San Antonio, October.

Chou, C. (1991). Cockpit Task Management Error: Design Issue fro Intelligent Pilot-vehicle Interfaces, Ph.D. Thesis, Dept. of Industrial and Manufacturing Engineering, Oregon State University.

Chou, C. and Funk, K. (1990). Management of Multiple Tasks: cockpit task management errors, *Proceeding of the 1990 IEEE International Conference on Systems, Man, and Cybernetics*, Los Angeles, 4-7 November, pp. 470-474.

Chou, C. and Funk, K. (1993). Cockpit task management errors in a simulated flight operation, in *Proceedings of the Seventh International Symposium on Aviation Psychology*, Columbus, OH, 26-29 April 1993.

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

Degani, A., and Wiener, E.L. (1990) The human factors of flight-deck checklist: The Normal Checklist (*NASA Contract Report 177549*), Moffett Field, CA: NASA Ames Research Center.

Degani, A., and Wiener, E.L. (1991) Philosophy, Policies, and Procedures: The Three P's of Flight-Deck Operation, *The Sixth International Symposium on Aviation Psychology*, April 29- May 2, Columbus, Ohio.

Degani,A. and Palmer, E. (1991) Electronic Checklist: Evaluation of Two Levels of Automation. Technical Report, Moffett Field, CA: NASA Ames Research Center.

Funk, K. (1991), Cockpit Task Management: Preliminary Definitions, Normative Theory, Error Taxonomy, and Design Recommendations, The International Journal of Aviation Psychology, vol. 1, no.1, pp.271-285.

Funk, K. & Kim, J. (1995), "Agent-based Aids to Facilitate Cockpit Task Management", *IEEE Transaction of Systems, Man, and Cybernetics* 1995.

Funk, K. & Lind,J. (1992) "Agent-Based Pilot-Vehicle Interface: Concept and Prototype" *IEEE Transaction of Systems, Man, and Cybernetics* 17 April

Funk, K. & McCoy, B. (1996) "A functional model of flightdeck agenda management", *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*, Sep. 2-6, 1996, Philadelphia, PA USA.

Geddes, N.D. (1985) "Intent Inferencing Using Script and Plans" *Proceeding of the First Annual aerospace Applications of Artificial Intelligence Conference*, Dayton, OH. 160-172.

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

Geddes,N.D. & Hammer, J.M,.(1991). Automatic Display Management Using Dynamic Plans and Events. *Information Working Meeting*, NASA Langely, 16-7, July.

Gerlach,M., Prevot, T., Ruckdeschel, W., Wittig, T., and Onken, R. (1995). Evaluation of Intelligent On-Board Pilot Assistance In In-flight Field Trials. *IFAC Man-Machine System '95*, June 27-29, 1995, Cambridge, MA.

Hammer, J. (1984) "An Intelligent Flight Management Aid for Procedure Execution " *I.E.E.E Transaction On Systems, Man, And Cybernetics*, SMC-14, No,6 885-888

Hawkins, F.H. (1993) Human Factors in Flight (2nd ed.) (edited by Harry W. Orlady) Vermont: Ashgate Publishing Company.

Hapeshi, L.C. and Jones, D.M. (1989). The ergonomics of automatic speech recognition interfaces. *International Review of Ergonomics*, 2, 251-290.

Hoshstrasser (1991) "The Design of an Object_Oriented Intent Inferencer for Integrated Systems. " *Digital Avionics Systems Conference*. Oct. 14-17, L.A.

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

Jones, P.M., MitchelL, C.M., and Rubin, K.S. (1988) Intent Inferencing with a model-based operator's associate. *Proceeding of the Sixth Symposium on Empirical Foundation of Information and Software Science.*

Jones , P.M., MitchelL, C.M., and Rubin, K.S. (1990) Validation of intent inferencing by a model -based operator's associate. *International Journal of Man Machine Studies*, 33, 177-202.

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

Lea, W.A. and Dahl, D. (1996). Designing User-Satisfying Voice Interfaces for Spoken Language Systems. *Proceedings of American Voice Input/Output Society*, San Jose, CA, Sep 10-12.

Madhavan, D. (1994) *Cockpit Task Management Errors: An ASRS Incident Report Study*. Unpublished master's thesis, Oregon State University.

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

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

Nagel, D.C. (1988) Human Error in Aviation Operations in Earl L. Wiener and D.C. Nagel, Human Factors in Aviation, San Diago: Academic Press, Inc., pp. 263-303.

National Transportation Safety Board (NTSB) (1988) *Northwest Airlines DC-9-82, N312RC, Detroit Metropolitan Wayne County Airport, Romulus, Michigan, August 16, 1987.* (aircraft accident report NTSB/AAR-88/05) Washington, DC:NTSB

NTSB (1989) *Delta Airlines Boeing 737-200 N473DA, Dallas-Fort Worth International Airport, Texas, August 31,*1988, (aircraft accident report NTSB/AAR-89/04) Washington, DC:NTSB.

NTSB (1990) *USAir,INC., Boeing 737-400 N416US, Laguardia Airport, Flushing, New York, September 20, 1989,* (aircraft accident report NTSB/AAR-90/03) Washington, DC:NTSB

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

Rasmussen, J. (1986) Information Processing and Human Machine Interaction: An Approach to Cognitive engineering. New York: North-Holland.

Reason, J., (1990) *Human Error*. New York: Cambridge University Press.

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

Roth, E.M., Bennett, K.B. and Woods, D.D. (1987) Human Interaction with an intelligent machine. *International Journal of Man-Machine Studies*, 27, 479-525.

Rouse, W,B. (1981) "Human-Computer Interaction in the Control of Dynamic Systems", *Computing Surveys* Vol. 13 No.1 March.

Rouse, W.B., Geddes, N.D., & Curry, R.E. (1987) "An Architecture for Intelligent Interface: Outline of an approach to supporting operators of complex systems", *Human Computer Interaction*, 3(2)..

Rouse, W.B., Geddes,N.D., & Hammer, J.M. (1990, March). Computer-aided fighter pilots. *IEEE Spectrum*, 27. 38-41.

Rubin, K.S., Jones, P.M. & Mitchell, C.M. (1988) *I.E.E.E Transaction On Systems, Man, And Cybernetics*, SMC-18, No,4 618-637.

Sarter, N.B. and Woods, D.D. (1992). Pilot Interaction with cockpit automation: Operational Experiences with the Flight Management System, *International Journal of Aviation Psychology*, 2(4), 303-322.

Sears, R.L. (1986) *A new look at accident contributions and the implication of operational and training procedures*. Unpublished report. Boeing Commercial airline Company.

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

Smith, W.G. (1994), A good game plan , *Technology Report* in Verbex Voice System brochure, 1994.

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

Wickens, C.D. (1992), Engineering Psychology and Human Performance (2nd ed.), New York: HarperCollins Publishers Inc.

Wickens, C.D. and Liu, Y. (1988). Codes and modalities for multiple resources: A success and some qualification. *Human Factors*, 30, 599-616.

Williamson, D.T., Barry, T.P. and Liggett, K.K. (1996). Flight Test Results Of ITT VRS-1290 IN NASA OV-10. *Proceedings of American Voice Input/Output Society*, San Jose, CA, Sep 10-12.

Zachary, W., Ross, L. (1991) Enhanced Human-Computer Interaction Through the Use of Embedded COGNET Models. *Proceedings of the Human Factors Society 35th Annual.*

APPENDICES

Appendix A. Simulator Manual Flight Normal Procedure

OSU 037 Plane Simulator

(Manual Flight Normal Procedure)

Simulator Manual Flight Normal Procedure

START PROCEDURE

Assume all procedures before TAKEOFF were already accomplished

Headphone .... SET
Covert GCM .... OPEN
Overt GCM .... OPEN
Agenda Manager Display .... CHECK
Flight Path MAP .... CHECK
Start Clearance .... OBTAIN

Read back ATC Command slowly and accurately. And check new Altitude Goal on Agenda Manager Display. Repeat to the ATC Altitude command if not declared.

OSU 037, Eugene Ground. Cleared to Portland via Eugene 4 departure, Corvallis, Victor 495, Newberg direct. ***Maintain 9000.*** *Expect 14000 5 minutes after departure.*

TAKEOFF PROCEDURE

VR Speed .... CONFIRM
Park Brake .... RELEASE

PILOT FLYING	PILOT NOT FLYING
Advance thrust lever to full throttle slowly.	
Verify thrust lever advance.	
Verify 80 knots.	Call out "80 knots"
Monitor airspeed and rotate at VR	Call out "V1" At VR, call out "ROTATE"
Call for "GEAR UP" and position landing gear lever up when positive rate of climb established.	

AFTER TAKEOFF PROCEDURE

Manual Flaps are not available.

PILOT FLYING	PILOT NOT FLYING
Monitor airspeed and Maintain V2 + 15 knots. Monitor altitude.	
Contact Departure and follow ATC instructions. *Read back ATC commands slowly and accurately.* *Check new Heading goal and Speed Goal on Agenda*	

Manager Display. Repeat to read back if no bell sound. Verify CVO VOR within range. Monitor direct CVO

CLIMB AND CRUISE PROCEDURE

Climb and cruise speed may be informed from ATC.

PILOT FLYING	PILOT NOT FLYING
Monitor climb speed and altitude. *If climb speed is not informed from ATC ,* *maintain initial climb speed 210 knots* Contact Seattle Center and follow their instruction. *Read back ATC commands slowly and accurately.* *Check new Altitude goal and Speed Goal on Agenda* *Manager Display. Repeat to read back if no bell sound.* After cruise altitude, maintain cruise speed. *If cruise speed is not informed from ATC ,* *maintain cruise speed 280 knots.* Verify UBG VOR within range. At top of descent point, Seattle Center contact for clearance and then follow instruction.	

APPROACH PROCEDURE

Manual Flaps are not available.

PILOT FLYING	PILOT NOT FLYING
Check the route from MAP. Contact Portland Approach. Follow their instruction. *Read back ATC commands slowly and accurately.* *Check new Altitude goal and Speed Goal on Agenda* *Manager Display. Repeat to read back if no bell sound* Verify PDX VOR within range. And Monitor direct PDX	

END OF FLIGHT

No landing is required.

Subjective Workload Measure Procedure

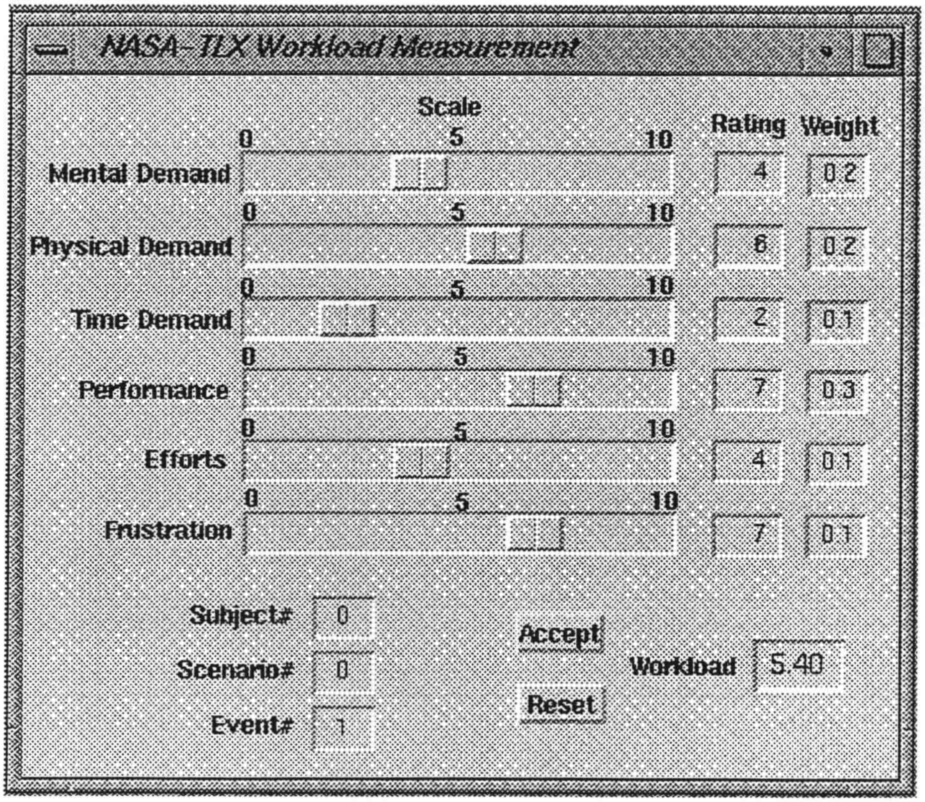

Figure. NASA TLX Scale Interface Display

- Experimenter will ask you to stop the flying and he will freeze the process for you, and then you are asked to measure your workload. You are required to rate your scale on six categories simply moving slide bar to the position you rated. The scale range is between 0 and 10. 0 means low or easy and 10 means high or hard. The six scales are defined as follows (Wickins 1994):

Mental Demand How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Endpoints: *Low/High*

Physical Demand How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Endpoints: *Low/High*

Time Demand How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic? Endpoints: *Low/High*

Performance How successful do you think you were accomplishing the goals of the task set by experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals? Endpoints: *Perfect/Failure*

Efforts How hard did you have to work (mentally and physically) to accomplish your level of performance? Endpoints: *Low/High*

Frustration How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task? Endpoints: *Low/High*

After rating the scale of each level, press *Accept* button. After that, experimenter will resume the process for you. And then continue to fly.

Appendix B. Overt GCM Training Procedure

Overt GCM Training Procedure

Overt GCM Training Procedure

Follow this procedure to Train the Recognizer to recognize your voice:

1) **Find a quite place for first-pass training**
   For the first phrase of Application Training, set up your system in the most quiet area you can have, where only the sound of your voice can be heard. Make a room with no fans or equipment running.

2) **Check your equipment**
   (see *Verbex Installation Manual* for further information)

3) **Install a recognizer file**
   Load a recognizer file from the host computer, must be transferred to the Recognizer's internal memory. Type the following at the command prompt, and press ENTER.

   vload -or overt_tr.grm

   When the transfer is complete, the following message will appears

   Operation Succeeded.

4) **Set up the Recognizer for Training.**
   Run the VTRAIN program by typing:

   vtrain

   When you are asked, enter your name and today's date.

5) **Position the microphone.**
   Here are some general guidelines to follow to get the best results from headset:

   * Try to male sure that the microphone will not shift and move while you are working.

   * Try to position the microphone about a thumb's width away from the corner of your mouth, out of the mainstream of your voice.

   * Be sure that your lips, check or mustache will not rub against the microphone.

   * Try to keep the face of the microphone near your lips, directed away from other sounds. This will help shield the microphone from noise.

6) **Make the recognizer sample the noise around you.**
   The recognizer will sample the noise around you. This sampling process is called calibration. You will hear three beeps in the earphone then the Recognizer will calibrate. When calibration is complete, another beep will be heard.

   Be sure to remain silent during calibration. This is **extremely** important--it affects the quality of all further training.

7) **Perform First-Pass Training**

   a) Press *SPACE*

The recognizer picks one of the phrases in the scripts in the recognizer file and breaks it down into its individual words. Then the recognizer displays the first of these words:

Say: maintain

b) Say the word.
When it hears you speak, the recognizer responds with:

Press SPACE to Confirm, R To Reject.
<Esc> to Quit. (1/32) <103>

The number between the "<" and the ">" shows how loudly the Recognizer thinks you said the word. Softest is <-128>, loudest is <127>. Whenever you speak into the recognizer, try to keep this number between <40> and <120>.

The fraction in parenthesis () shows the number of WORDS in the recognizer file that the recognizer has heard you speak so far.

c) If you think you mispronounced the word you spoke, or you said it too softly, too loudly, or too late, press R to "reject" the word. I asks for the word again.

d) When the recognizer has heard all of the words in the phrase it picked from the script, it displays the phrase for you to speak:

Say:
maintain heading 290

e) Say the phrase
The recognizer attempts to understand what you say. It displays what it believes it has heard below the phrase:

Say:
maintain heading 290
maintain heading 290

If the phrase it hears does not match the phrase it asked for, the recognizer displays the following:

Press SPACE to Force training,
R to Reject, <ESC> to Quit. (1/187)

The fraction in parentheses () shows how many PHRASES the Recognizer has heard you speak so far. In this example, the trainer is on the first of 187 phrases.

If you mispronounced the phrase you spoke, or you coughed or cleared your throat, etc., press R to reject the phrase. Otherwise, press SPACE to tell the recognizer that you spoke correctly.

When all words have been spoken several times in various phrases, First-Pass Training ends the following message:

Training of Vocabulary 1 Complete
Press SPACE to continue

If you press SPACE, the following appears:

Perform another pass of Training?
Yes No

Press Y to perform more First-Pass Training (Overall training quality may improve).

You may have a break before you perform another training.

8) **Save your voice sample**

If you press N at the "Perform another pass of training?" prompt, the recognizer responds with the following message:

Save Pattern?
Yes No

Press Y to instruct the recognizer to keep the voice samples it got during First-Pass Training. After the recognizer saves the samples, the TRAIN menu reappears.

9) **Save your voice file to disk**

Using the VSAVE software utility, type the following, and press ENTER.

vsave -ov *vfilename*

VSAVE will display following messages as it transfers the data.

Operation Succeeded.

10) **Install the recognizer file and voice file**

Using VLOAD software, you can load the files to a Recognizer's memory and send the response data to the Agenda Manager. Recognition file, at this moment, is different for the one for training since its display response should be different. (i.e., "climb to 9000" in training vs. "a1 9000"). So type the following:

vload -orv overt *vfilename*   " vfilename is voice file"

Then VLOAD will display following messages as it transfers the data.

Operation Succeeded.

Tips for Training

1. **Relax** You are training the recognizer to work the way you want it to, NOT vice versa.

2. **Speak in your normal voice** as you will speak when you actually use the recognizer to recognize.

3. **Speak consistently** - try not to change much the way you say a word each time you say it.

4. **Remember** that you will be able to retrain any word or words you need to change.

5. **Resist the temptation to resist** what you have said unless you are sure you said phrase wrong. Each tine you force a word or phrase you are training the recognizer. If you reject the phrase, you are only training yourself.

And most of all:

6. **Do not worry that you will do something wrong**. You cannot damage the recognizer by pressing a wrong key, etc.

Testing Recognition

You should test recognition performance after creating your Voice file, to see if you need to perform any additional application training.

1) Put a headset on and adjust it as illustrated in *Overt GCM Training Procedure.*

2) Type the following, and press ENTER.

    vtrain -s

3) In the MAIN menu, press T for "Train".

4) In the TRAIN menu, press R for "Rec-Test".

5) Follow the instructions and answer the questions on your screen until the following message appear:

    LISTENING... Press E To Record A
    Mis-Recognition, <ESC> to Quit Test.

6) Speak a complete, valid utterance for your application into the microphone.

7) The recognizer will display what it believes it has heard from you.

8) Each time the recognizer does not seem to hear you speak a phrase, or responds with or displays a different phrase from that which you spoke, press E for error. Otherwise, press SPACE to continue with Rec-Test.

Appendix C. Scenario

Scenario

OSU 037 FMC Plane Eugene-Portland Scenario

(Version 1)
08/02/96

Initial Conditions: Lat/Long -- 44.07/123.13
Alt -- 365 MSL Freq -- 121.7 Mhz
Spd -- 0 KIAS Hdg -- 210 MAG (Runway 21)
Flaps -- n/a

EVENTS	ATC	OSU 037
OSU 037, ready for clearance delivery	OSU 037, Eugene Ground. Cleared to Portland via Eugene 4 departure, Corvallis, Victor 495, Newberg, direct. Maintain 9000. Expect 14000 5 minutes after departure.	Cleared to Portland via Eugene 4 departure, Corvallis, Victor 495, Newberg, direct. ***Maintain 9000***, expecting 14000 5 minutes after departure.
OSU 037, ready for take-off.		Eugene Tower, OSU 037, ready for take-off on runway 21.
	OSU 037, Eugene Tower, runway 21, cleared for take-off.	Cleared for take-off on ***Runway 21***.
OSU 037 parking brake off , full throttle, and takes-off. OSU 037 landing gear up		
OSU 037 at 1400	OSU 037 contact Departure	Eugene Departure, OSU 037, out of 1500 for 9000.
OSU 037 at 2500	OSU 037, Eugene Departure, radar contact. Turn right heading 290.	
		OSU 037, ***turn right heading 290***
OSU 037 at 3400	OSU 037, turn right heading 330. When able proceed direct Corvallis. Maintain speed 240. Alt. indicates 3500.	OSU 037, ***turn right heading 330, direct Corvallis, maintain speed 240.***

EVENTS	ATC	OSU 037
OSU 037 at 8000	OSU 037, contact Seattle Center 125.8.	Seattle Center, OSU 037 out of 8000 for 9000
	OSU 037, Seattle Center, climb and maintain 14000.	OSU 037 ***climb and maintain 14000***.
OSU 037 at CVO	OSU 037, Seattle Center, maintain current heading 352	***Maintain current heading 352***
OSU 037 at 14000	OSU 037, maintain speed 280	OSU 037, ***maintain speed 280.***
OSU 037 at TOD	OSU 037, reduce speed to 240, descend at pilot's discretion. Maintain 10000, direct UBG	OSU 037, ***reduce speed to 240, descend to 10000. direct UBG***
OSU 037 at UBG	OSU 037, turn left heading 334	OSU 037, ***turn left heading 334.***
OSU 037 at 10000.	OSU 037, contact Portland Approach 133.0.	Portland Approach, OSU 037, at 10000 with BRAVO.
	OSU 037, turn right heading 360	OSU 037, ***turn right heading 360***
	OSU 037, Portland Approach, descend and maintain 7000. Reduce speed to 210.	OSU 037, ***descend to 7000*** and ***reduce speed to 210***
OSU 037 at 7000	OSU 037, turn right heading 010 and descend and maintain 4000.	OSU 037, ***turn right heading 010, descend and maintain 4000***
OSU 037 at 4000	OSU 037, 5 miles from YORKY, turn right heading 070. Maintain 3000 until established on the localizer. Cleared ILS runway 10R for approach.	OSU 037, ***turn right heading 070, maintain 3000*** til established. Cleared ILS runway 10R

Appendix D. Overt GCM Grammar File

Overt GCM Grammar File

```
;overt.grm                 updated by Woochang Cha 09/13/96
;grammar file for displaying subject voice for developing overt goal communication method
;simplified  by not-enrolling unnecessary word based scenario atcscen2

#application demo

#vocabulary Communication
 #grammar main_grammar
  #main_structure
   . CMD
  #end_structure
  #structure .1-2
   1
   2
  #end_structure
  #structure .0-2
   0
   .1-2
  #end_structure
  #structure .1-3
   .1-2
   3
  #end_structure
  #structure .0-3
   0
   .1-3
  #end_structure
  #structure .1-5
   .1-3
   4
   5
  #end_structure
  #structure .0-5
  0
  .1-5
  #end_structure
  #structure .1-9
   .1-5
   6
   7
   8
   9
  #end_structure
  #structure .0-9
   0
   .1-9
  #end_structure
  #structure .ALT_NUM
   .1-9
   10
   .1-3 .0-9
```

```
#end_structure
#structure .ALT
  .ALT_NUM thousand
#end_structure
#structure .ALTITUDE_CMD
   maintain altitude/ .ALT
   .ALT_CMD .ALT
   .ALT_CMD and maintain .ALT
#end_structure
#structure .ALT_CMD
  climb
  descend
#end_structure
#structure .SPD_NUM
  .1-3 .0-9 0
  .1-3 niner 0
#end_structure
#structure .SPEED_CMD
  .SPEED speed .SPD_NUM
#end_structure
#structure .SPEED
  maintain
  reduce
  increase
#end_structure
#structure .HEADING_CMD
  turn .LR heading .HDG_NUM
  maintain/ heading .HDG_NUM
  runway .0-3 .0-5
#end_structure
#structure .LR
  left
  right
#end_structure
#structure .HDG_NUM
  .0-2 .0-9 .0-9
  3 .0-5 .0-9
#end_structure
#structure .WAYPOINT_CMD
  direct .WAYPOINT
#end_structure
#structure .WAYPOINT
  CVO
  UBG
  PDX
#end_structure
#structure .CMD
  .ALTITUDE_CMD
  .HEADING_CMD
  .SPEED_CMD
  .WAYPOINT_CMD
#end_structure

#end_grammar
```

```
#end_vocabulary

#display_response
  #translations
   #initiator ""
   #separator ""
   #terminator "\n\r"
   niner "9"
   thousand "000"
  #templates
   climb .ALT > "a1 " .ALT
   climb and maintain .ALT > "a1 " .ALT
   descend .ALT > "a2 " .ALT
   descend and maintain .ALT > "a2 " .ALT
   maintain .ALT > "a3 " .ALT
   maintain altitude .ALT > "a3 " .ALT
   maintain speed .SPD_NUM > "b1 " .SPD_NUM
   reduce speed .SPD_NUM > "b1 " .SPD_NUM
   increase speed .SPD_NUM > "b1 " .SPD_NUM
   turn left heading .HDG_NUM > "c1 " .HDG_NUM
   turn right heading .HDG_NUM > "c2 " .HDG_NUM
   maintain heading .HDG_NUM > "c3 " .HDG_NUM
   heading .HDG_NUM > "c3 " .HDG_NUM
   runway .0-3 .0-5 > "c3 " .0-3 .0-5
   direct .WAYPOINT > "d1 " .WAYPOINT

#end_application
```

```
;overt_tr.grm                updated by Woochang Cha 09/13/96
;grammer file for training subject voice for developing overt goal communication method
;simplified  by not-enrolling unnecessary word based scenario atcscen2

#application demo

#vocabulary Communication
 #grammar main_grammar
  #main_structure
   . CMD
  #end_structure
  #structure .1-2
   1
   2
  #end_structure
  #structure .0-2
   0
   .1-2
  #end_structure
  #structure .1-3
   .1-2
   3
  #end_structure
  #structure .0-3
   0
   .1-3
  #end_structure
  #structure .1-5
   .1-3
   4
   5
  #end_structure
  #structure .0-5
  0
  .1-5
  #end_structure
  #structure .1-9
   .1-5
   6
   7
   8
   9
  #end_structure
  #structure .0-9
   0
   .1-9
  #end_structure
  #structure .ALT_NUM
   .1-9
   10
   .1-3 .0-9
```

```
#end_structure
#structure .ALT
 .ALT_NUM thousand
#end_structure
#structure .ALTITUDE_CMD
  maintain altitude/ .ALT
  .ALT_CMD .ALT
  .ALT_CMD and maintain .ALT
#end_structure
#structure .ALT_CMD
 climb
 descend
#end_structure
#structure .SPD_NUM
 .1-3 .0-9 0
 .1-3 niner 0
#end_structure
#structure .SPEED_CMD
 .SPEED speed .SPD_NUM
#end_structure
#structure .SPEED
 maintain
 reduce
 increase
#end_structure
#structure .HEADING_CMD
 turn .LR heading .HDG_NUM
 maintain/ heading .HDG_NUM
 runway .0-3 .0-5
#end_structure
#structure .LR
 left
 right
#end_structure
#structure .HDG_NUM
 .0-2 .0-9 .0-9
 3 .0-5 .0-9
#end_structure
#structure .WAYPOINT_CMD
 direct .WAYPOINT
#end_structure
#structure .WAYPOINT
 CVO
 UBG
 PDX
#end_structure
#structure .CMD
 .ALTITUDE_CMD
 .HEADING_CMD
 .SPEED_CMD
 .WAYPOINT_CMD
#end_structure

#end_grammar
```

```
#end_vocabulary

#display_response
  #translations
   #initiator ""
   #separator " "
   #terminator "\n\r"
     CVO "Corvallis"
     UBG "Newburg"
     PDX "Portland"
     0 "zero"
     1 "one"
     2 "two"
     3 "three"
     4 "four"
     5 "five"
     6 "six"
     7 "seven"
     8 "eight"
     9 "niner"
     10 "ten"
  #templates
   .CMD > \WORD_LIST

#end_application
```