Project #: E-24-613
Cost share #: E-24-317
Rev #: 5
Center #: 10/24-6-R7012-0A0
Center shr #: 10/22-1-F7012-0A0
OCA file #: 
Work type: RES
Contract#: NCC 2-675
Mod #: 8/25/94 LETTER
Contract entity: GTRC
Subprojects #: Y
Prime #: 
Main project #: 
Project unit: ISYE Unit code: 02.010.124
Project director(s): MITCHELL C M ISYE (404)894-4321
GOVINDARAJ T ISYE (404)-
Sponsor/division names: NASA / AMES RESEARCH CTR, CA
Sponsor/division codes: 105 / 006
Award period: 900701 to 940930 (performance) 940930 (reports)
Sponsor amount New this change Total to date
Contract value 0.00 344,563.00
Funded 0.00 344,563.00
Cost sharing amount 30,000.00
Does subcontracting plan apply ?: N
Title: OPERATOR MODELING IN COMMERCIAL AVIATION: COGNITIVE MODELS, INTELLIGENT....

PROJECT ADMINISTRATION DATA

OCA contact: Anita D. Rowland 894-4820
Sponsor technical contact Sponsor issuing office
DR. EVERETT A. PALMER MS. BARBARA HASTINGS
(415)604-6073 (415)604-5802
NASA AMES RESEARCH CENTER NASA AMES RESEARCH CENTER
AVIATION SYSTEMS RESEARCH BRANCH, UNIVERSITY AFFAIRS BRANCH,
239-21 241-1
MOFFETT FIELD, CA 94035-1000 MOFFETT FIELD, CA 94035-1000
Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : N/A N/A supplemental sheet
Equipment title vests with: Sponsor GIT X
SPONSOR'S PRIOR APPROVAL REQUIRED IF >$5,000 AND NOT IN PROPOSAL.
Administrative comments - SPONSOR'S 8/25/94 LETTER "MOD" APPROVES A NO-COST EXTENSION.
GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 06/12/95

Project No. E-24-613__________ Center No. 10/24-6-R7012-0A0_

Project Director MITCHELL C M__________ School/Lab ISYE__________

Sponsor NASA/AMES RESEARCH CTR, CA________________________________

Contract/Grant No. NCC 2-675______________ Contract Entity GTRC

Prime Contract No. ________________

Title OPERATOR MODELING IN COMMERCIAL AVIATION: COGNITIVE MODELS, INTELLIGENT..

Effective Completion Date 940930 (Performance) 940930 (Reports)

<table>
<thead>
<tr>
<th>Closeout Actions Required</th>
<th>Date Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Invoice or Copy of Final Invoice</td>
<td>Y  950518</td>
</tr>
<tr>
<td>Final Report of Inventions and/or Subcontracts</td>
<td>Y  ________</td>
</tr>
<tr>
<td>Government Property Inventory &amp; Related Certificate</td>
<td>Y  ________</td>
</tr>
<tr>
<td>Classified Material Certificate</td>
<td>N  ________</td>
</tr>
<tr>
<td>Release and Assignment</td>
<td>N  ________</td>
</tr>
<tr>
<td>Other</td>
<td>N  ________</td>
</tr>
</tbody>
</table>

Comments: _______________________________________________________

Subproject Under Main Project No. ________________________________

Continues Project No. __________________________________________

Distribution Required:

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Director</td>
<td>Y</td>
</tr>
<tr>
<td>Administrative Network Representative</td>
<td>Y</td>
</tr>
<tr>
<td>GTRI Accounting/Grants and Contracts</td>
<td>Y</td>
</tr>
<tr>
<td>Procurement/Supply Services</td>
<td>Y</td>
</tr>
<tr>
<td>Research Property Management</td>
<td>Y</td>
</tr>
<tr>
<td>Research Security Services</td>
<td>N</td>
</tr>
<tr>
<td>Reports Coordinator (OCA)</td>
<td>Y</td>
</tr>
<tr>
<td>GTRC</td>
<td>Y</td>
</tr>
<tr>
<td>Project File</td>
<td>Y</td>
</tr>
<tr>
<td>Other</td>
<td>N</td>
</tr>
</tbody>
</table>

NOTE: Final Patent Questionnaire sent to PDPI.
GEORGIA INSTITUTE OF TECHNOLOGY  
OFFICE OF CONTRACT ADMINISTRATION  

NOTICE OF PROJECT CLOSEOUT (SUBPROJECTS)  

---  

Closeout Notice Date 06/12/95  

Project No. E-24-613  

Project Director MITCHELL C M  

Center No. 10/24-6-R7012-0A0_  

School/Lab ISYE  

Sponsor NASA/AMES RESEARCH CTR, CA  

---  

<table>
<thead>
<tr>
<th>Project #</th>
<th>E-24-697</th>
<th>PD MITCHELL C M</th>
<th>Unit 02.010.124</th>
<th>T</th>
<th>OTH #</th>
<th>NCC 2-675</th>
<th>MOD#</th>
<th>8/25/94 LTR</th>
<th>ISYE *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctr #</td>
<td>10/24-6-R7012-0A1</td>
<td>Main proj # E-24-613</td>
<td>OCA CO ADR</td>
<td></td>
<td>Sponsor</td>
<td>NASA /AMES RESEARCH CTR, CA</td>
<td>105/006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATOR MODELING</td>
<td>IN</td>
<td>Start 920701</td>
<td>End 940930</td>
<td>Funded 92,185.00</td>
<td>Contract 92,185.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---  

LEGEND  

1. * indicates the project is a subproject.  
2. I indicates the project is active and being updated.  
3. A indicates the project is currently active.  
4. T indicates the project has been terminated.  
5. R indicates a terminated project that is being modified.
Operator Modeling in Commercial Aviation:  
Cognitive Models, Intelligent Displays, and Pilot's Assistants

NASA grant (NASA Ames control number NCC 2-675; 90-55)

Semi-Annual Report:

July 1, 1990 through December 31, 1990

T. Govindaraj  
C. M. Mitchell
Center for Human-Machine Systems Research  
School of Industrial and Systems Engineering  
Georgia Institute of Technology  
Atlanta, GA 30332-0205  
(404) 894-2300; fax (404) 894-2301  
E24-613
**Background**

One of the goals of the national Aviation Safety/Automation program is to address the issue of human-centered automation in the cockpit. Human centered automation is automation that, in the cockpit, enhances or assists the crew rather than replacing them. The Georgia Tech research program focuses on this general theme, with emphasis on designing a computer-based pilot's assistant and intelligent (i.e., context-sensitive) displays. In particular, the aids and displays will be designed to enhance the crew's situational awareness of the current state of the automated flight systems and to assist the crew in coordinating the autoflight system resources.

The goal of a computer-based assistant is to enhance the pilot situational awareness of the active autoflight modes and their interactions and to facilitate quick recovery from unsafe conditions. To be effective, the aid must incorporate a detailed model of each pilot's function in the cockpit resource management system. The model must also contain knowledge of flight system states, knowledge of situational interactions between automated systems, and knowledge about the use of information sources and the allocation of cockpit tasks.

The operator function model (OFM) provides one framework to structure knowledge for the 'intelligent aid'. Knowledge about nominal flight operations structured into a cockpit OFM could endow the aid with the power to infer pilot intent and predict the consequences of pilot decisions regarding the use of automation in a dynamic environment.

Like previous related work on the operator function model, we are developing a model that includes representation of current crew and systems activities, and, given this knowledge, the model can construct a representation of the current state of the crew and aircraft, and, given this knowledge can predict an 'envelope' of expected crew activities. In a computational form, OFMspert (operator function model expert system), an intelligent aid that builds an evolving hypothesis of current operator intentions, and given that knowledge, can offer timely advice, reminders, and, when asked by the crew, assume control of lower level subsystems.

**Progress To-Date**

The first six months of this grant saw the design of an operator function model for the pilot-flying and pilot-not-flying on the Boeing 727 flight deck. Given the OFM, an OFMspert was built and its interpretations were successfully compared to those of two Boeing 727
pilots. The work was interesting for two reasons: first, several OFMspert features had to be extended or added (e.g., 'phase' as in phase of flight); and second, the OFMspert for a 727 was dependent on manual entry of extensive 727 verbal interactions of the crew, air traffic control, etc. The latter, although highlighting the lack of practicality of an OFMspert for the 727, suggests that OFMspert was potentially very appropriate for the more digitized interaction in glass-cockpit airplane.

The 727 OFM and OFMspert work was presented at the 1990 International Conference on Systems, Man and Cybernetics (Appendix A), as a preliminary technical report (Appendix B), and is being prepared as Ms. Smith Verfurth's M.S. thesis. We expect this research to result in additional papers, including a Center for Human-Machines Systems Research technical report, conference proceedings paper, and a refereed journal publication. The real-time OFMspert is available on either a Sun3 or Mac II running Allegro Common Lisp.

In addition, to Serena S. Verfurth's research, this grant is supporting Todd Callantine's doctoral research and M.S. theses for two new graduate students, Jim Williams (a private pilot) and Ed Crowther. Todd's research, Appendix C, explores the extension of the OFMspert architecture to more explicitly address the needs of the glass flight deck.

The two new graduate students, Jim and Ed, initially focused on understanding the goal of human-centered aviation. To this end we developed a real-time interactive simulation of the 737 interfaces to autoflight systems, including the 737 the mode control panel, ADI, HSI, and CDU. The system includes a flight model and we are working on the map-mode for the HSI. Although this is early in their studies, we expect these students to participate in the on-going Georgia Tech research program by pursuing 'intelligent' flight deck displays (Jim) and computer-based tutoring for 767 autoflight systems. We expect the first year of this grant to define several research directions: OFMspert, training, aiding and intelligent displays for the glass cockpit.

To further explore computers on the flight deck, we developed an informal relationship with United's training program and a somewhat more formal relationship with Delta's Aviation Human Factors program.
Appendix A


Appendix B

Smith, S. C., Govindaraj, T. and Mitchell, C. M. Operator Modeling in Civil Aviation. technical report draft, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, 1990

Appendix C

Callantine, T. J. Computer-Supported Mode Awareness and Planning in Glass Cockpit Aircraft, technical report draft, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, 1990
Operator Modeling in Commercial Aviation:
Cognitive Models, Intelligent Displays, and Pilot's Assistants

NASA grant (NASA Ames control number NCC 2-675; 90-55)

Semi-Annual Reports: #3, 4 & 5

July 1, 1991 through December 31, 1991
January 1, 1992 through July 15, 1992
July 16, 1991 through December 31, 1992

T. Govindaraj
C. M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2300; fax (404) 894-2301
E24-613
Background

One of the goals of the national Aviation Safety/Automation program is to address the issue of human-centered automation in the cockpit. Human centered automation is automation that, in the cockpit, enhances or assists the crew rather than replacing them. The Georgia Tech research program focuses on this general theme, with emphasis on designing a computer-based pilot's assistant and intelligent (i.e., context-sensitive) displays. In particular, the aids and displays will be designed to enhance the crew’s situational awareness of the current state of the automated flight systems and to assist the crew in coordinating the autoflight system resources.

The goal of a computer-based assistant is to enhance the pilot situational awareness of the active autoflight modes and their interactions and to facilitate quick recovery from unsafe conditions. To be effective, the aid must incorporate a detailed model of each pilot’s function in the cockpit resource management system. The model must also contain knowledge of flight system states, knowledge of situational interactions between automated systems, and knowledge about the use of information sources and the allocation of cockpit tasks.

The operator function model (OFM) provides one framework to structure knowledge for the 'intelligent aid'. Knowledge about nominal flight operations structured into a cockpit OFM could endow the aid with the power to infer pilot intent and predict the consequences of pilot decisions regarding the use of automation in a dynamic environment.

Like previous related work on the operator function model, we are developing a model that includes representation of current crew and systems activities, and, given this knowledge, the model can construct a representation of the current state of the crew and aircraft, and, given this knowledge can predict an 'envelope' of expected crew activities. In a computational form, OFMspert (operator function model expert system), an intelligent aid that builds an evolving hypothesis of current operator intentions, and given that knowledge, can offer timely advice, reminders, and, when asked by the crew, assume control of lower level subsystems.
Research Orientation

Problem: Effective Use of Computer Technology and Automation on the Commercial Flight Deck--specifically, automation applications that enhance the effectiveness of pilots in the glass cockpit.

Objectives: Four related projects comprise this research program. Taken together they define a spectrum of research investigating human-centered application of automation on the flight deck. First, construction of a part-task simulator of the major autoflight controls and displays on the flight deck of a Boeing 757/767; this project will support empirical evaluation of the research on the related projects. Second, development of an intent inferencing structure based on the operator function model (OFM) to provide the intelligence for a context-sensitive pilot's assistant; the goal is to provide a computer-based pilot's assistant that is as flexible and 'understanding' as a second officer. Third, development of an intelligent tutor, and, potentially, an embedded on-line aid to facilitate pilot manipulation, monitoring, and reprogramming of the autoflight systems; this project focuses specifically on vertical profile. Fourth, and finally, the development of avionics displays that exploit state-of-the-art graphics and computer-based visualization technology. Specifically, this project entails development of a 3-D Flight Path Perspective display that facilitates continuous pilot awareness of the flight management system (FMS) and auto flight director system (AFDS) status; to enhance spatial awareness, this display will overlay the FMS flight path on a 3-D artificial horizon, incorporating AFDS and FMS queues both alphanumerics and graphical representations.

Progress To Date

Glass Cockpit Crew Intent Inferencer: Research Progress and Future Work (Todd J. Callantine)

Research at Georgia Tech toward an intent inferencing system for glass cockpit aircrews has shown that the use of a representation of the aircraft's limiting operating envelope is useful in the inferencing process. Using the scheme described below, the intent inferencer can produce a description of the crew's current activities from simulator data read from a file. Modifications to the intent inferencer will enable it to track ACFS crew activities in real time. These modifications are outlined after the limiting operating envelope and inferencing process are discussed.
The limiting operating envelope is a sequence of constraints, imposed by the flight plan, ATC clearances, and operational guidelines, that defines the desired state of the aircraft along the flight path. The currently binding set of constraints is termed the active limit state of the limiting operating envelope. By comparing the actual autoflight system state, flight management system (FMS) state, and aircraft state to the active limit state, an event-based description of the current flight situation can be generated.

The inferencing process uses this description as a flexible means of indexing activities in a hierarchical crew model. The model itself is functional decomposition of activities; each phase of flight is decomposed into crew functions, which are in turn decomposed into subfunctions, autoflight mode selections, tasks, subtasks, and, at the lowest level, actions. Each activity in the model has an associated set of conditions for determining the status of the activity based on the occurrence of a particular event or events. By noting the statuses and changes in the statuses of activities in the model (e.g., "active," "pending," "done"), a useful description of the crew's current activities is produced.

Several modifications are needed to fully exploit the above inferencing scheme. Already in progress are enhancements to the conditions used in the model. A broader set of status indicators, including "assumed-active" and "assumed-done" (for describing undetectable activities), and conditions that specify one, some, or all of the events must occur before the status may change have been introduced. Also in progress are enhancements that allow enroute changes to the limiting operating envelope due to ATC directives. Finally, FMS state representations must be enriched to reflect the variety of ways in which the FMS can be used by the crew. This modification must be congruent with the capabilities of the ACFS FMS.

Flight Management System Research Test Bed (Ed Crowther and Jim Williams)

The Boeing 757 flight instrument simulator is designed to facilitate research relating to the interaction between the pilot and the aircraft's flight management system (FMS). A comprehensive flight model of the B757 is used to drive high fidelity displays of the primary navigation instrumentation. An FAA navaid data base is used to simulate commercial flight routes. Interactive features of the simulator allows for flight from takeoff to full stop landing. The simulator provides the pilot with all available FMS and Flight Director (FD) navigation functions.
The simulator incorporates a flexible design to provide for reconfiguration and modification. This modular structure allows for adaptation of the simulator to specific research interests, such as the Vertical Navigation Tutor, 3-D FMS Primary Flight Display, and Pilot Intent Inferencer projects.

Flight Path Perspective 3-D Primary Flight Display (Jim Williams)

Traditional primary flight displays (PFDs) provide minimum spatial orientation to the pilot with regards to flight management system (FMS) programmed routes. The goal of this research is to integrate FMS and flight director (FD) routes along with basic aircraft flight parameters into a single display. The display format follows modern PFDs while incorporating a unique perspective format or "tunnel in the sky" coupled to the existing FMS and FD.

Flight Management System Vertical Navigation Tutor (Ed Crowther)

Vertical navigation capabilities of the Flight Management System (FMS) in modern "glass-cockpit" aircraft are often underutilized or misused by pilots. This can be attributed at least in part to an inadequate understanding by pilots of how the FMS interprets and executes a flight plan which they have entered. This project combines a unique vertical profile display with the B757 simulator. The display provides an otherwise unavailable visual representation of FMS and other vertical navigation modes of the aircraft. A control architecture is embedded into the system to allow for the creation of routine flights which the tutor uses as lessons that address key training issues. The tutor controls flight scenarios which help the student pilot explore the content of the FMS vertical profile, FMS execution of that profile through use of the VNAV function, interaction between FMS and other vertical navigation modes, and the use of FMS vertical navigation by the pilot for the completion of various in-flight maneuvers.
Operator Modeling in Commercial Aviation: Cognitive Models, Intelligent Displays, and Pilot's Assistants

NASA grant (NASA Ames control number NCC 2-675; 90-55)

Semi-Annual Reports: #6

January 1, 1993 through July 15, 1993
E-24-613; E-24-697

T. Govindaraj
C. M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2300; fax (404) 894-2301
E24-613
Background

One of the goals of the national Aviation Safety/Automation program is to address the issue of human-centered automation in the cockpit. Human centered automation is automation that, in the cockpit, enhances or assists the crew rather than replacing them. The Georgia Tech research program focuses on this general theme, with emphasis on designing a computer-based pilot's assistant and intelligent (i.e., context-sensitive) displays. In particular, the aids and displays will be designed to enhance the crew's situational awareness of the current state of the automated flight systems and to assist the crew in coordinating the autoflight system resources.

The goal of a computer-based assistant is to enhance the pilot situational awareness of the active autoflight modes and their interactions and to facilitate quick recovery from unsafe conditions. To be effective, the aid must incorporate a detailed model of each pilot's function in the cockpit resource management system. The model must also contain knowledge of flight system states, knowledge of situational interactions between automated systems, and knowledge about the use of information sources and the allocation of cockpit tasks.

The operator function model (OFM) provides one framework to structure knowledge for the 'intelligent aid'. Knowledge about nominal flight operations structured into a cockpit OFM could endow the aid with the power to infer pilot intent and predict the consequences of pilot decisions regarding the use of automation in a dynamic environment.

Like previous related work on the operator function model, we are developing a model that includes representation of current crew and systems activities, and, given this knowledge, the model can construct a representation of the current state of the crew and aircraft, and, given this knowledge can predict an 'envelope' of expected crew activities. In a computational form, OFMspert (operator function model expert system), an intelligent aid that builds an evolving hypothesis of current operator intentions, and given that knowledge, can offer timely advice, reminders, and, when asked by the crew, assume control of lower level subsystems.
Research Orientation

Problem: Effective Use of Computer Technology and Automation on the Commercial Flight Deck—specifically, automation applications that enhance the effectiveness of pilots in the glass cockpit.

Objectives: Four related projects comprise this research program. Taken together they define a spectrum of research investigating human-centered application of automation on the flight deck. First, construction of a part-task simulator of the major autoflight controls and displays on the flight deck of a Boeing 757/767; this project will support empirical evaluation of the research on the related projects. Second, development of an intent inferencing structure based on the operator function model (OFM) to provide the intelligence for a context-sensitive pilot's assistant; the goal is to provide a computer-based pilot's assistant that is as flexible and 'understanding' as a second officer. Third, development of an intelligent tutor, and, potentially, an embedded on-line aid to facilitate pilot manipulation, monitoring, and reprogramming of the autoflight systems; this project focuses specifically on vertical profile. Fourth, and finally, the development of avionics displays that exploit state-of-the-art graphics and computer-based visualization technology. Specifically, this project entails development of a 3-D Flight Path Perspective display that facilitates continuous pilot awareness of the flight management system (FMS) and auto flight director system (AFDS) status; to enhance spatial awareness, this display will overlay the FMS flight path on a 3-D artificial horizon, incorporating AFDS and FMS queues both alphanumeric and graphical representations.

Progress To Date

Glass Cockpit Crew Intent Inference: Research Progress and Future Work (Todd J. Callantine)

Research at Georgia Tech toward an intent inferencing system for glass cockpit aircrews has shown that the use of a representation of the aircraft's limiting operating envelope is useful in the inferencing process. Using the scheme described below, the intent inferencer can produce a description of the crew's current activities from simulator data read from a file. Modifications to the intent inferencer will enable it to track ACFS crew activities in real time. These modifications are outlined after the limiting operating envelope and inferencing process are discussed.
The limiting operating envelope is a sequence of constraints, imposed by the flight plan, ATC clearances, and operational guidelines, that defines the desired state of the aircraft along the flight path. The currently binding set of constraints is termed the active limit state of the limiting operating envelope. By comparing the actual autoflight system state, flight management system (FMS) state, and aircraft state to the active limit state, an event-based description of the current flight situation can be generated.

The inferencing process uses this description as a flexible means of indexing activities in a hierarchical crew model. The model itself is functional decomposition of activities; each phase of flight is decomposed into crew functions, which are in turn decomposed into subfunctions, autoflight mode selections, tasks, subtasks, and, at the lowest level, actions. Each activity in the model has an associated set of conditions for determining the status of the activity based on the occurrence of a particular event or events. By noting the status and changes in the activities of the model (e.g., "active," "pending," "done"), a useful description of the crew's current activities is produced.

Several modifications are needed to fully exploit the above inferencing scheme. Already in progress are enhancements to the conditions used in the model. A broader set of status indicators, including "assumed-active" and "assumed-done" (for describing undetectable activities), and conditions that specify one, some, or all of the events must occur before the status may change have been introduced. Also in progress are enhancements that allow enroute changes to the limiting operating envelope due to ATC directives. Finally, FMS state representations must be enriched to reflect the variety of ways in which the FMS can be used by the crew. This modification must be congruent with the capabilities of the ACFS FMS.

A demonstration of this system was given at NASA Ames in the spring.

**Flight Management System Research Test Bed (Ed Crowther and Jim Williams)**

GT-EFIRT (Georgia Tech Electronic Flight Instrument Research Tool) is a part-task Boeing 757 flight instrument simulator. GT-EFIRT is designed to facilitate research relating to the interaction between the pilot and the aircraft's flight management system (FMS). A comprehensive flight model of the B757 is used to drive high fidelity displays of the primary navigation instrumentation. An FAA navaid data base is used to simulate commercial flight routes. Interactive features of the simulator allows for flight from
takeoff to full stop landing. The simulator provides the pilot with all available FMS and Flight Director (FD) navigation functions.

The simulator incorporates a flexible design to provide for reconfiguration and modification. This modular structure allows for adaptation of the simulator to specific research interests, such as the Vertical Navigation Tutor, 3-D FMS Primary Flight Display, and Pilot Intent Inferencer projects.

This period saw the completion of GT-EFIRT. It is now a stable tool and is in subsequent stages of revision to add enhancements. We are currently adding the ability to hand fly the simulation using either joy stick or control yoke inputs. The flight and control systems models are unique to GT-EFIRT. This is due to the initial design of the models to always be in full auto pilot, which precluded the need to understand or model manual input to flight control surfaces. As such, manual input is modeled via gain changes in the auto pilot. For example, a roll induced heading change replaces the equivalent heading select input on the MCP.

Other enhancements completed during this period include:

- Full control over the collection of state variables and experimenter notes at any time during the session. Each entry is time stamped.
- State variable capture "snapshot" and storage with subsequent retrieval and loading as the current state array. This includes aircraft position and orientation.
- Aircraft slew in all axis during simulation pause to evaluate terrain maps.

**Flight Management System Vertical Navigation Tutor (Ed Crowther)**

Vertical navigation capabilities of the Flight Management System (FMS) in modern "glass-cockpit" aircraft are often underutilized or misused by pilots. This can be attributed at least in part to an inadequate understanding by pilots of how the FMS interprets and executes a flight plan which they have entered. This project combines a unique vertical profile display with the B757 simulator. The display provides an otherwise unavailable visual representation of FMS and other vertical navigation modes of the aircraft. A control architecture is embedded into the system to allow for the creation of routine flights which the tutor uses as lessons that address key training issues. The tutor
controls flight scenarios which help the student pilot explore the content of the FMS vertical profile, FMS execution of that profile through use of the VNAV function, interaction between FMS and other vertical navigation modes, and the use of FMS vertical navigation by the pilot for the completion of various in-flight maneuvers.

This period saw the completion of the VNAV Tutor and initial evaluation. The evaluation consisted of training 5 Delta transition pilots (with little or no previous 'glass' experience). A demonstration of the VNAV Tutor was given at the NASA Ames AS/A meeting in August.

**Flight Path Perspective 3-D Primary Flight Display with Terrain Information (Jim Williams)**

EFIS equipped aircraft such as the B757/767 and MD11 are replacing the aging commercial fleet. The EFIS suite of avionics, to date, still includes conventional ground proximity warning systems (GPWS). An effort is underway at Georgia Tech to integrate enhanced ground proximity warning into the primary flight display (PFD). The primary focus of this research is to evaluate controlled flight into terrain (CFIT). The approach utilized is based upon many years of research into flight path perspective displays or "tunnel in the sky" displays. By coupling flight path data based upon either manual input or FMS commanded modes into a 3-D (perspective) format, the flight crew would be afforded a significant increase in warning time for impending terrain. The terrain would be represented on the display in an "out the nose" format with the flight path superimposed. This integrated format replaces the traditional attitude director indicator (ADI) and has been named the Spatial Situation Indicator (SSI).

The SSI display is driven by GTEFIRT. All other display and control requirements are met using this tool. We have completed the design of the basic instrument including preliminary runs with a B757/767 rated pilot. Significant work still exist in the design of the terrain data base, specifically with regards to the required resolution. The remaining effort will be placed on speed improvements. GTEFIRT coupled to the SSI is currently only capable of 5 frames per second display update rates.
Operator Modeling in Commercial Aviation: 
Cognitive Models, Intelligent Displays, and Pilot's Assistants

NASA grant (NASA Ames control number NCC 2-675; 90-55)

Semi-Annual Reports: #7
July 15, 1993 through January 1, 1994
E-24-613; E-24-697

T. Govindaraj
C. M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2300; fax (404) 894-2301
E24-613
This report summarizes the work carried on as part of the no-cost extension of this grant. It specifically addresses the final design of the spatial situation indicator display (SSI). Appendix A contains a thorough description of the proposed displays together with the proposed experimental design for evaluating them. The paper in Appendix A is a conference paper presented at 1993 IEEE International Conference on System, Man, and Cybernetics in October describing this research. Appendix B contains an overview of the software architecture.

This period also saw the evaluation of the proposed displays. Eighteen Delta Air Lines pilots within the B757/767 fleet evaluated three displays: the traditional displays in the B757/767, a conventional primary flight display modeled after the Rockwell-Collins Pro-Line series PFD, and the spatial situation indicator (SSI) which included terrain and predictive flight path information. The experimental design consisted of three scenarios which corresponded with documented CFIT or near CFIT incidence. Each display was flown against the three scenarios by individual pilots with two replications of the experiment. Pilots who flew the SSI formatted display were provided a post-session questionnaire to assist in qualitative analysis of the display.

Preliminary results based upon the questionnaire are very positive. Quantitative data analysis is currently underway.
Appendix A
Effects of Integrated Flight Path and Terrain Displays on Controlled Flight into Terrain

James A. Williams and Christine M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta GA 30332-0205
williams, cm@chmsr.isye.gatech.edu

ABSTRACT This paper describes research on the development of a prototype integrated display for terrain avoidance in the terminal phase of flight. The display combines auto pilot-coupled flight path data overlaid onto a terrain grid map using a primary flight display format.

I. INTRODUCTION

Spatial awareness in a multi-mode flight environment requires a rich set of both static and dynamic visual cues integrated into a consistent, easy to understand display. Current electronic flight instrument systems have performed this integration through the use of a mode control panel (MCP) for auto pilot interaction, a flight management system (FMS) and control display unit (CDU) for flight planning and routing, a horizontal situation indicator (HSI) and an attitude director indicator (ADI) for display of aircraft orientation. The pilot scan of these current state-of-the-art instruments must shift from one display to the next with no consistency. Newer light commercial aircraft such as the Beech Starship with Rockwell-Collins avionics are moving towards integrated flight displays using multi-mode CRTs in a primary flight display (PFD) configuration. These wide format CRT displays lend themselves to increased spatial economy thereby avoiding display clutter and increased visual workload. The research efforts have concentrated on the PFD format and the inclusion of a three dimensional attitude display to re-enforce the pilot's model of both lateral and vertical navigation in near-terrain situations. This new display format is referred to as the Spatial Situation Indicator (SSI). Specific emphasis has been placed on the terminal phase of flight with terrain modeling in the vicinity of the departing and destination airport. Experimental analysis and simulation support for this effort utilizes the Georgia Tech Electronic Flight Instrument Research Tool (GTEFIRT). GTEFIRT is a part-task flight simulator specifically designed to study aircraft display design and single pilot interaction. The simulator, using commercially available graphics workstations, replicates a high level of fidelity the Electronic Flight Instrument Systems (EFIS), Flight Management Computer (FMC) and Auto Flight Director System (AFDS) of the Boeing 757/767 aircraft. The simulator can be configured to present information using conventional looking B757/767 displays or next generation Primary Flight Displays (PFD) such as found on the Beech Starship and MD-11.

The simulator provides high fidelity representations of the interfaces and responses of the auto flight and instrumentation systems while remaining low-cost, rapidly reconfigurable, and portable. Its object oriented design allows new displays to be prototyped quickly and evaluated through flight scenarios with complete data logging of pilot and systems performance. All navigation related aural and visual alerts/warnings are modeled including the ground proximity warning system (GPWS).

A. System Configuration

As shown schematically in Figure 1, the baseline version of GTEFIRT utilizes two computers and three monitors. The right two monitors are touch sensitive and all pilot interaction can be performed using touch input. The workstations are connected via a local area network (LAN). A Sun SPARC 2 workstation with a GS graphics accelerator card drives the left most monitor. This monitor and CPU are specifically design for the 3-D flight path and terrain displays associated with the PFD. The UNIX operating system and the Sun OpenLook Toolkit provide flexibility in allocating displays among the CPUs and monitors. The typical configuration is identified in the figure but any combination of monitors and display windows can be requested. For example, the simulation support panel which is used by the researcher can be allocated to a workstation anywhere on the LAN.

Air traffic control (ATC) interaction is carried out by the researcher with real-time event logging in the data collection file.

B. Software Design
Modular design and rapid reconfiguration were the driving factors in designing the structure of GTEFIRT. An object oriented architecture was chosen which is implemented not only in the source programming language but also in the selection of Sun PHIGS+ as the graphics support language and the Sun OpenLook Toolkit for window management. The underlying simulation is based on a three degrees of freedom point mass model of the B757. This model provides sufficient fidelity of aircraft dynamics since no hand flying is implemented. As such, pitch and thrust are the driving forces with no modeling of aircraft control surfaces. The control loops for the auto flight system can operate in several different modes, ranging from simple altitude and heading hold to a full lateral and vertical path guidance based on FMC programmed routes. Localizer and glideslope tracking modes can be engaged for final approach and provide for complete category III full stop landings including the flare maneuver. Computational speed has been enhanced by parallel processing the simulation task across two CPUs. The flight model, FMC and AFDS are allocated to one CPU while the navigation and moving map are allocated to the other. The two CPUs are synchronized via message traffic on the LAN.

C. Data Collection

Extensive data collection capabilities are built into GTEFIRT for both system and pilot monitoring. A data log is maintained for each session whose contents are selectable by the researcher. Events which can be monitored include pilot input, auto pilot state changes, aircraft dynamics, and aircraft related alerts (e.g., flap and gear warnings). Events are recorded with a time stamp.

III. PRIMARY FLIGHT DISPLAY INTEGRATED DESIGN OBJECTIVES AND REQUIREMENTS

The implementation of advanced technology into today's "glass cockpit" has allowed modern commercial aircraft to "auto land" in zero visibility conditions at specified airports. This heads down flying eliminates the static and dynamic out-the-window visual cues normally present in good to marginal weather conditions. An approach to simulate these visual cues is the "out-the-nose" integrated display which replaces the conventional ADI with a virtual representation of the horizon and surrounding terrain. A variety of symbology can be overlaid onto the terrain map to include: man-made obstructions, a perspective flight path, and radio navigation aids. It has been shown in previous work by Ellis (1987) and Roscoe (1980) that a flight path display furnished with adequate predictor information enhances the pilot's ability to fly arbitrary curved approaches. With the advent of Global Positioning System (GPS) navigation a curved approach through terrain is now possible. Of principal concern to this research was the need to alert the crew to future terrain clearance problems rather than relying on conventional ground proximity warning systems (GPWS) which look at current radar altitude. The pathway (or tunnel) display has been tested elaborately in the past e.g., Dorighi (1992), and shown a significant potential for increased pilot situational awareness.

Perspective flight path displays can be categorized into two areas, aircraft fixed or ground fixed. The auto pilot mode of operation of greatest concern was the semi-automatic mode which includes heading select, vertical speed, heading hold, altitude hold, and flight level change. In the near-terrain scenarios modeled here the crew is normally flying in one of these sub-modes rather in the fully automatic modes of vertical (VNAV) or lateral (LNAV) navigation. As such, the flight path becomes aircraft fixed and moves forward across the ground with aircraft motion. This aircraft (or pilot) body reference translates into an egocentric display format allowing more rapid processing of terrain clearance information.

In addition to the perspective flight path cue of terrain clearance, a gross signal of terrain closure was desired to provide additional information to the conventional GPWS. Terrain modeling for this effort required a data base which represented the general terrain picture in the vicinity of the departure and destination airports. These data needed to present sufficient detail so that steering information could be derived from the display to aid the pilot in terrain avoidance.

IV. SPATIAL SITUATION INDICATOR DESIGN

The design of the PFD was motivated by a need to present an auto pilot-coupled flight path overlaid onto a terrain map making both obstructions and terrain perceptually available. A hypothesis present by Aretz (1982) is that a pilot maintains navigational awareness through the constant alignment of two frames of reference: the ego centered reference frame (ERF) and world centered reference frame (WRF). The projected flight path represents the ERF and the terrain map and underlying grid represent the WRF. Terrain avoidance now becomes a cognitive process by which the pilot maintains a congruent relationship between the ERF and WRF. The use of this format entailed a choice of many specific parameters for the projection; for example, the field of view (synthetic focal length), position and direction of view, and vertical scale expansion. Since there is little theoretical guidance for making these choices, they were made primarily on discussion with other research team members and airline pilots flying the preliminary model. The final projection chosen was therefore somewhat ad hoc.
Figure 1. GTEFIRT System Configuration

Figure 2. Terrain clearance "whiskers" and flight path.
A. Perspective Flight Path

The unique design of the flight path presents predicted position and terrain clearance information for up to 75 seconds ahead of the aircraft. Projection of the flight path is based on a “fast time” modeling technique described by Grunwald (1985). Traditional flight paths use the ‘tunnel in the sky’ approach which present no reference to the ground elevation e.g., Grunwald (1982). The technique developed for this research utilized roll stabilized vertical lines “whiskers” positioned at 15 second intervals out to 75 seconds. Figure 2 illustrates the virtual “whiskers” and flight path. The whiskers are displayed in pairs of equal distant widths so that in steady level flight a perspective path is projected. The whiskers are color coded using green and yellow. The green lower portion extends from the predicted aircraft altitude at that interval to the terrain below. Its length therefore is a direct representation of the terrain clearance at that point in the aircraft’s path given no changes in aircraft dynamics. The upper portion of the whisker is fixed at 2000 feet in length and extends upward from the predicted aircraft altitude. The tops of each whisker or connected to provide a visual flow which is highly useful in discerning the flight path during moderate to steep turns. The whisker is “consumed” by the terrain as the aircraft moves along its path. A gross cue of terrain clearance is simply the presence of green on the vertical whisker. If green is visible, there is clearance ahead, if no green is visible, then impact with the terrain is eminent. A finer cue of terrain clearance is available by comparing the lengths of the green and yellow segments. For example, if the green lower segment is half the length of the upper yellow segment, then 1000 feet of terrain clearance is available at that predicted position.

B. Terrain Grid

In situations where the finer detail of actual terrain clearance is less important, the terrain is color coded dynamically. The mechanism for determining the color coding is based upon aircraft predicted height and terrain spot elevations. The color coding uses dark green for safe terrain and dark red for dangerous terrain. Gouraud shading techniques are used to create a smooth transition. The terrain grid is comprised of a triangular mesh with each triangle having sides of 2 nautical miles (NM). The terrain map is 28 NM wide and 12 NM wide. This technique works well since each scenario is only concerned with the approach to landing phase of flight. The outer edge of the terrain map has each spot elevation set to sea level. The remaining 28 spot elevation are hand coded into computer readable files which are loaded with each scenario. The terrain grid is overlaid on a sea-level grid for all other areas of the flight plan. Man-made obstructions such as radio towers are hand coded and read with the terrain files. Information for building the terrain and obstruction files is obtained for the approach plates for each runway in the scenario.

V. EXPERIMENTAL APPARATUS

Mode awareness as it relates to aviation safety is well documented, e.g., Sarter (1992). Pilot performance is predictable is most situations that have been studied using perspective flight displays. In collaboration with NASA, mode and spatial awareness during controlled flight into terrain (CFIT) was chosen as the focus for this research. Pilot performance and mode awareness has not been subject to extensive study in this environment.

An extensive review of the reference literature with specific emphasis on Bateman (1991) identified three candidate near CFIT situations to model. These three case studies represent types of errors which can be introduced into the approach by the pilot flying in an EFIS equipped aircraft. These cases are a) Mode Reversion, b) Incorrect MCP Altitude, and c) Improper Barometric Altitude.

The experimental evaluation uses the GITEFIRT simulation in two modes of operation: the baseline B757/767 simulation using conventional displays, and with the Primary Flight Display. The PFD can be configured with or without visual cues for AFDS or FMS altitudes, the pathway in the sky coupled to the controlling auto pilot, and terrain.

The objectives of the experimental program are:
1) To evaluate the performance of the pilot in near CFIT situations with conventional B757/767 displays.
2) To evaluate the performance of the pilot in near CFIT situations with a conventional PFD similar to that found on MD-11 aircraft.
3) To evaluate the performance of the pilot in near CFIT situations with pathway in the sky and terrain maps integrated into the PFD.

A. Design

The experiment will focus on the approach phase of the programmed flight route. In each case the aircraft will be prepositioned as identified in the individual case study. Aircraft state variables (the state vector) will be loaded so that the aircraft will start the scenario with parameters set as indicated in each case study.

Attention diverting tasks will be implemented to match as closely as possible the scenarios as they are described by Bateman. ATC communications will be implemented using simple voice communications without supporting electronic intercoms. The experimenter will carry out the traffic controller (ATC) communications.

The goal of the experiments is to track improvement in (reduction of) CFIT or near CFIT situations. A clock will be maintained throughout the experiment which is simulation task driven. Response time of the pilot for corrective action will be monitored as well as MCP inputs. A time stamp of each event and input will be recorded for each subject.
Each subject will have between 30 and 60 minutes of flight experience with the simulation. Each mode of operation of the PFD will be demonstrated allowing subjects time to interact and familiarize themselves prior to the experimental run. It is anticipated that nine pilots will be selected. Each case study will be run with and without PFD auxiliary displays (tunnel, markers, etc.) and with the baseline simulation for a total of three display configurations. Each case comprises a scenario and two subordinate scenarios based on the original case have been developed giving a total of nine scenarios. The statistical model used is a Greco Latin Squares with two repetitions. Therefore, each subject will participate in nine sessions.

Case 1: Mode Reversion

**Aircraft:** B737-300  
**Location:** Ontario, California  
**Date:** 16 November, 1990

**Description:** While on final approach to ILS runway 26L, the aircraft inadvertently descended to less than 390 feet, some 1400 feet below the outer marker crossing altitude of 2800 feet. A MSAW occurred, the aircraft advised, and a missed approach was made. There was no GPWS warning.

**Remarks:** It had been reported that the aircraft was cleared for ILS 26L near PETIS, then suddenly recleared for VOR runway 26R with AFS engaged for approach mode. The aircraft was almost established on the VOR approach when it was recleared back to the ILS. THE AFDS did not couple to glideslope and reverted to a vertical speed mode. There is significant terrain left of the localizer course, and the aircraft fortunately missed the terrain and towers.

**Scenario 1:** The simulation will start with aircraft position enroute PETIS on the 318° radial at 4500 feet. Clearance will be granted for the ILS 26L approach prior to reaching PETIS. It is assumed that the pilot will select APP mode on the AFDS. Upon LOC capture at PETIS and the turn toward a course of 256°, the aircraft will be recleared to VOR runway 26R. This will occur prior to establishing the 256° heading. The pilot should deselect APP mode by disengaging the AFDS then re-engage by selecting any of the CMD autopilots. This places the aircraft in heading hold and V/S mode. This should be followed by pilot selection of HEADING SELECT and a V/S of 700 fpm. Upon the aircraft reaching 270° the pilot will be recleared back to the ILS 26L approach. The pilot will select APP mode again. The simulation will respond with LOC capture but will fail (not capture) the GS. The experiment will continue until either the GPWS alerts or the pilot initiates corrective action.

**Case 2:** Incorrect MCP Altitude

**Aircraft:** A320  
**Location:** San Diego, California  
**Date:** June, 1990

**Description:** Aircraft was enroute SWATT (versus VYDDA) under HEADING SELECT mode on the FMGS. V/S mode was engaged and the aircraft was descending to 3700 feet (100' above the approach plate indicated altitude). The MCP was inadvertently set to 2700 feet.

**Remarks:** Incident based upon sketchy data provided by flight crew. The aircraft may have been in L - V NAV but this is not likely since the MCP altitude is usually verified as part of the FMS data entry and to miss both settings would be highly unlikely.

**Scenario 2:** The simulation will start with aircraft position enroute SWATT from DOUGA at an altitude of 5000 feet. The AFDS will be engaged in HEADING SELECT and V/S mode with the MCP altitude set to 2700. The simulated terrain enroute SWATT will consist of a mountain peak at 2900'. The experiment will continue until either the GPWS alerts or the pilot initiates corrective action for the MCP altitude.

**Case 3:** Improper Barometric Altimeter Setting

**Aircraft:** B737-200  
**Location:** Boise, Idaho  
**Date:** 17 February, 1990

**Description:** While on an ILS approach to runway 10R, the aircraft inadvertently was 1000 feet below the FAF with both altimeters indicating 4100 feet. On the missed approach, it was discovered that the altimeter setting were incorrect.
Scenario 3: The simulation will start with aircraft position enroute USTIK at an altitude of 5000 feet indicated. The actual altitude is 800 feet lower. The AFDS will be engaged in APP mode with the LOC captured and GS well above aircraft position. The experiment will continue until either the GPWS alerts or the pilot initiates corrective action for the improper barometric altimeter setting.

B. Subjects

All subjects chosen for the experiment are Delta Airlines B757/767 type rated instructor pilots. Each pilot will be paid $200 for participation.

VI. CONCLUSION

The major goal of this research is to investigate the role of flight path and terrain representations as components of an “out-the-nose” virtual display. Historically, the terrain avoidance logic developed by airlines has emphasized maneuvers in the vertical profile. With the advent of the spatial situation indicator, it will be of interest to observe if maneuvers in the horizontal plane will become part of standard terrain escape procedures.

VII. ACKNOWLEDGMENTS

This research is supported by NASA Ames Research Center (Everett Palmer, Technical Monitor). The authors would very much like to thank all the pilots and ground training personnel at Delta Air Lines who provided extensive training and documentation on the B757/767.

VIII. REFERENCES


Bateman, D., Some Examples of Collision with Terrain Accidents and Incidents Where the Autopilot, Flight Director, or Flight management System was Thought to be in use by the Pilot, unpublished report to NASA, Sundstrand Data Control, January 1991.


Figure 4. Spatial Situation Indicator.
APPENDIX B

GT-EFIRT (SSI) SOURCE CODE ARCHITECTURE

GT-EFIRT

SSI_baseline

B757_baseline

SSI

hsi

flights

B757

B757.c
global_data.c
comms.c
set_color.c
control_panel.c
gpwss.c
sound.c
afds_fms.c
ssi.c
mcp.c
ssi_map.c
flight_model.c
cdu.c
cdu_page_calls.c
fms_support.c

global_data.h
Default_data
Makefile

glideslope.au
pull_up.au
screech.au
sink_rate.au
minimums.au
terrain.au
warning_short.au
cautaion.au
cautaion_alert.au
caustria.au
flaps.au
gear.au
libaudio.a
custom_termio.h
libaudio.h
audio_device.h
audio_errno.h
audio_hdr.h

KATL-KBHM
KBHM-KATL
KBOS-KLGA
KLGA-KBOS
S1
S2
S3

S1
cdu_pages
fms_data

dep_obstacles
dep_terrain
dest_obstacles
dest_terrain
snapshot

fms_data

approach1
approach2
pre_flight_plan

approach1
approach2
pre_flight_plan

approach1
approach2
pre_flight_plan
afds_fms.c

void afds_fms(delta_time)
void altitude_capture(altitude)
void check_mcp_alt_cap()
void check_vnav_alt_cap()

B757.c

Notify_value destroy_func(client, status)
void main(argc, argv)
Notify_value one_sec_timer()
void pause_clock()
Notify_value quarter_second_timer()
void start_clock()
Notify_value timer_loop()
int wst_best_dbuf_mode(wst, xconnid)

cdu.c

static void build_cdu_base()
Notify_value cdu_destroy_func(client, status)
void cdu_event_proc(window, event)
void cdu_open_workstation(screen)
static void set_view_rep()

cdu_page_calls.c

void build_arrival_selected_page(approach)
void build_cdu_pages()
void CLR_button_selected()
void descend_now_function_selected()
void EXEC_button_selected()
Ppoint get_text_pts(text_field)
void line_1L_selected()
void line_1R_selected()
void line_2L_selected()
void line_2R_selected()
void line_3L_selected()
void line_4L_selected()
void line_5L_selected()
void line_6L_selected()
void line_6R_selected()
void next_page_selected()
void page_selected(selected_page)
void prev_page_selected()
void read_CDU_data(cdu_page_file)
int selected_page;
void select_transition()
void set_CDU_message(message_number)
void set_flight_scenario()
void update_CLB_PAGE()
void update_CRZ_PAGE()
void update_DES_PAGE()
void update_LEGS_active_wpt()
void update_LEGS_PAGE()
void update_PROG_PAGE1()
void update_PROG_PAGE2()

comms.c
void close_external_nav()
int init_comms(fds, sin_port_offset)
void init_comms()
void init_ext_nav_port()
void Notify_value recv_from_hsi(client, fd)
void send_flight_plan_name_to_hsi(header, file_name)
void send_position_to_hsi(header, data1, data2)
void send_to_hsi(header, data1, data2, data3)
void send_wpt_to_hsi(header, wpt_name, distance, eta)
void sim_sleep(sec, usec)
void transmit_nav(a_lat, a_long, track, speed)

control_panel.c
void build_flight_plan_list()
void clear_eicas(item, event)
void create_panel(same_server, root, server)
int data_file_selected(item, event)
void eicas_alert(message_id, status)
void eicas_repaint(canvas, pw, display, xid, x rects)
void file_menuProc(menu, menu_item)
void initialize(item, event)
void log_entry(log_text)
void Notify_value monitor_scroll(client, event, sbar, type)
void save_file_pushed(item, event)
void set_AFDS_path(item, val)
void set_alt_markers(item, val)
void set_baro_alt(item, val)
void set_brakes(item, val)
void set_clock(item, val)
void set_course(item, value, event)
void set_dh(item, val)
void set_ext_nav(item, event)
void set_flaps(item, val)
void set_flight(item, string, client_data, op, event)
void set_gear(item, val)
void set_gs_capture(item, val)
void set_monitor_metric(item, val)
void set_slew_incremen t(item, value, event)
void set_spoilers(item, value, event)
void set_terrain(item, val)
void set_thrott le(item, value, event)
void set_tms_1(item, val)
void set_tms_2(item, val)
void show_slew(item, event)
void shut_down(item, event)
void silence_alarms(item, val)
void slew_down(item, event)
void slew_east(item, event)
void slew_north(item, event)
void slew_return(item, event)
void  slew_roll_ccw( item, event )
void  slew_roll_cw( item, event )
void  slew_south( item, event )
void  slew_up( item, event )
void  slew_west( item, event )
void  slew_yaw_ccw( item, event )
void  slew_yaw_cw( item, event )
void  snapshot(item, event)
void  sound_volume( item, val )
void  test_sound(item, event)
void  tune_nav_receiver( item, val )

flight_model.c
void  flight_model(delta_time)
void  flight_path_predictor(advance_time)

fms_support.c
float  bearing_between_index(point_A, point_B)
double bearing_between_xy(x1, y1, x2, y2)
void  build_vnav_route()
void  check_gs_intercept()
void  check_lnav_intercept()
void  check_loc_intercept()
void  compute_vnav_event_dist()
void  compute_vtk_error()
int  delta_heading()
float  distance_between_points(point1, point2)
float  distance_to_track()
void  follow_gs()
void  follow_track()
float  geo_to_cart(geo_degrees)
int  get_active_wpt()
void  get_adjacent_wpt_indices()
void  get_approach_index()
float  get_eta_at_point(distance)
void  Get_FMS_route_info(flight_plan_file)
int  get_following_wpt(index)
float  get_vnav_RoC(speed)
float  get_vnav_speed()
int  LNAV_switch_wpt()
void  plot_ToC()
double point_range(x1,y1,x2,y2)
void  read_snapshot(scenario_directory)
float  seg_climb_accel_dist(start_altitude, end_altitude, start_speed, end_speed)
float  seg_descent_decel_dist(start_altitude, end_altitude, start_speed, end_speed)
void  start_speed, end_speed)
void  update_hsi_wpt()
void  update_ILS()
void  update_nav_DME()
double wpt_bearing(flag, index)
gpws.c
void  gpws()
mcp.c

void blank_mcp_speed()
void build_mcp()
void change_mcp_alt_hold()
void change_mcp_approach()
void change_mcp_center_ap()
void change_mcp_flch()
void change_mcp_hdg_hold()
void change_mcp_left_ap()
void change_mcp_inav()
void change_mcp_loc()
void change_mcp_n1()
void change_mcp_right_ap()
void change_mcp_speed()
void change_mcp_vnav()
void change_mcp_vs()
void init_pick_filter()
Notify_value mcp_destroy_func(client, status)
void mcp_event_proc(window, event)
void mcp_open_workstation(screen)
void show_mcp_vertical_speed(state)
void update_mcp_altitude()
void update_mcp_heading()
void update_mcp_speed()
void update_mcp_vertical_speed()

set_color.c

void set_color(colour_model)

sound.c

void close_sound()
double get_volume()
void init_sound()
void play_sound(file)
void repeat_sound()
void set_volume(volume)
Notify_value sigpoll_handler(client, sig, when)

ssi.c

static void build_adi_overlay()
static void build_alerts()
static void build_altitude_ladder()
static void build_compass_course_card()
static void build_fd_vee()
static void build_glideslope()
static void build_localizer()
static void build_pitch_ladder()
static void build_roll_ptr()
static void build_rose()
static void build_speed_ladder()
static void build_spt()
static void build_SSI_fms_alt_right()
static void build_SSI_mcp_alt_left()
static void build_SSI_mcp_alt_right()
static void build_text_indicators()
static void build_vertical_speed()
void build_vspeed_ind()
void clear_alert_boxes()
void init_view_mapping()
void minimums()
void rotate_rose()
void set_afds_es()
void set_afds_pma()
void set_afds_pme(alert_box)
void set_afds_rma()
void set_afds_rme(alert_box)
void set_altimeter()
void set_atme(alert_box)
void set_bug_speed()
void set_fd_kee()
void set_glideslope()
void set_heading_bug()
void set_localizer()
void set_nav_receiver()
void set_pitch_ladder()
void set_pred_speed()
void set_ref_speed(bug1, bug2, bug3, bug4, to_flag)
void set_rose_view_rep()
void set_speed()
void set_speed_view_rep()
void set_ssi_offset_view_rep()
static void set_ssi_window()
void set_support_indicators()
void set_vspeed_ind()
void set_vspeed_view_rep()
Notify_value SSI_destroy_func(client, status)
void ssi_event_proc(window, event)
void ssi_open_workstation(vis_depth, vis_class, colr_mode)
void update_SSI_fms_altitude()
void update_SSI_mcp_altitude()
void update_ssi_world_view_rep()

ssi_map.c

void build_AFDS_path()
void build_dep_runway()
void build_dep_terrain()
void build_dest_runway()
void build_dest_terrain()
void build_global_terrain()
void build_map()
void build_sky()
void build_smoke_trail()
void build_SSI_FMS_route()
float comp_terrain_alt(x_value, y_value)
Point3 Get_symbol_points(i)
void height_color_bias(dep_arpt, index1, index2)
void init_SSI_map()
void move_map()
void update_AFDS_path()
void update_grid()
void update_sky()
void updateTerrain()
Operator Modeling in Commercial Aviation:
Cognitive Models, Intelligent Displays, and Pilot's Assistants

NASA grant
(NASA Ames control number NCC 2-675; 90-55)

Final Report

July 1, 1990 through April 30, 1994

T. Govindaraj
C. M. Mitchell
Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2300; fax (404) 894-2301
E24-613
**Background**

One of the goals of the national Aviation Safety/Automation program is to address the issue of human-centered automation in the cockpit. Human-centered automation is automation that, in the cockpit, enhances or assists the crew rather than replacing them. The Georgia Tech research program focused on this general theme, with emphasis on designing a computer-based pilot’s assistant, intelligent (i.e., context-sensitive) displays, and an intelligent tutoring system for understanding and operating the autoflight system. In particular, the aids and displays were designed to enhance the crew’s situational awareness of the current state of the automated flight systems and to assist the crew in coordinating the autoflight system resources.

This grant funded four separate activities: an OFMspert to understand pilot navigation activities in a 727 class (i.e., 'non-glass') aircraft; an extension of OFMspert to understand mode control in a glass cockpit, Georgia Tech Crew Activity Tracking System (GT-CATS); the design of a training system to teach pilots about the vertical navigation portion of the flight management system--VNAV Tutor, and a proof-of-concept display, using existing display technology, to facilitate mode awareness, particularly in situations in which controlled flight into terrain (CFIT) is a potential.

**OMFspert for Navigation in the Non-Glass Cockpit (Serena Smith Verfurth)**

**Background**

OMFspert for the 727 was the first component of the research funded by this grant. The current grant built on research funded at Georgia Tech under a previous Ames grant. The previous grant helped to fund the development of OFMspert (operator function model expert system).

OMFspert is both a theory and model of the use of machine intelligence to aid the human operator. Essentially OFMspert builds a real-time model of operator activity in the context of unfolding system events--it attempts to understand operator(s) actions. Given this understanding, OFMspert can then offer reminders, advice, and assistance. OFMspert can be considered one instance of human-centered automation--automation that augments operator skills rather than replacing them.

The heart of an OFMspert is an operator function model (OFM). An OFM represents how a well-trained, well motivated operator coordinates control and monitoring activities in real time. It represents how an operator decomposes (i.e., extracts features from) and represents external events, acts those representations to create new information and produces behavior from the representations and processes. The operator function model is a hierarchical decomposition that is goal-oriented at the top and event/data-driven at the bottom.
An Operator Function Model of Boeing 727 Navigation

As the first test of its generalizability, OFMspert was applied to data from the B-727 flight simulator at NASA Ames Research Center. In a B-727 the crew performs both manual and cognitive actions while controlling flight. Manual actions include changing radio frequencies, adjusting autopilot settings, and maneuvering power using the throttles. Cognitive actions include monitoring crucial displays within the cockpit, as well as listening and responding to verbal commands and requests from both Air Traffic Control (ATC) and other crew members in the cockpit.

The B-727 OFM represents the joint pilot-autopilot control of the B-727, specifically the activities of the pilot-flying and the pilot-not-flying. While a three member crew flies the B-727, consisting of a pilot-flying, a pilot-not-flying, and a flight engineer, the role of the flight engineer is not included in the model. In the B-727, the flight engineer's role consists primarily of executing checklists and monitoring tasks. The flight engineer, therefore, is not involved directly in the control of flight. In more sophisticated civil aviation aircraft the role of the flight engineer has been eliminated. Therefore the role of the flight engineer is not included in the model.

The B-727 operator function model has five phases of flight: Cruise, Cruise Descent Transition, Descent, Descent Approach Transition, and Approach. Six functions comprise the function level of this OFM network. There are three pilot flying functions: Monitor/Modify Lateral Track, Monitor/Modify Vertical Profile, and Reconfigure/Check Aircraft. There are three PNF functions: Monitor Lateral Track, Perform ATC Communications, and Reconfigure/Check Aircraft. The function level, which is the highest level of description in the OFM, represents pilot function and purpose in the context of the entire pilot-aircraft system. For example, the functions in this OFM are of comparable priority and take place concurrently. The pilot-flying is involved in Monitoring/Modifying the Lateral Track as well as Monitoring/Modifying the Vertical Profile continuously during Cruise to Landing. In addition, during the Approach phase, the pilot-not-flying is also involved in decisions on when to Reconfigure and Check the Aircraft. The pilot flying from Cruise to Landing has the functions of Monitoring the Lateral Track, and Performing ATC Communications. Just as the pilot-flying, the pilot-not-flying also has the additional function of Reconfigure/Check the Aircraft during the Approach phase. Figure 1 depicts one piece of the OFM: Monitor/Modify Lateral Track for the pilot flying.

OFMspert for the B-727
Figure 2 depicts a snapshot of the OFMspert Backboard, ACTIN (actions interpreter), that performs the inferencing function. Top portion is the blackboard representation; the bottom is the state space which includes actual and desired state variables derived from the flight plan and ATC clearances.

Figures 2, 3, and 4 illustrate OFMspert’s operation. Depicted in Figure 2, at 509 seconds from the beginning of the flight scenario, the aircraft is more than 60 miles from Denver on the J56 radial at a cruising altitude of 31,000 feet. In Figure 2, the pilot-flying is maintaining an airspeed of 255 knots as requested by ATC. The pilot-flying’s navigation radio (number 1) is tuned to 112.8 MHz (Gill). The pilot-not-flying’s navigation radio is tuned to 117.0 (Denver). At 529 second, the pilot-flying announces top-of-descent. The ACTIN update is depicted in Figure 3. At 530 seconds, the pilot-flying retards the throttles to idle to descend the aircraft to 11,000 feet. The retard throttles action is posted on the lowest level of ACTIN, the actions level, and depicted in the figure with the circle labeled A. At 533 second the pilot-flying manipulates the rate-of-descent knob to adjust the pitch to maintain airspeed during descent. This action is connected to the change airspeed/pitch task node in Figure 3. In addition, the desired altitude has been updated to 11,000 feet (Circle B).

Validation
Two expert pilots were asked to indicate the intent of each action while observing the B727 OFMspert ACTIN interface. A total of 106 actions were reviewed by each pilot. The experts found only nine incorrect action connections (i.e., OFMspert inferences to be in error). Pilot reaction to OFMspert’s ability to correctly interpret actions was quite positive.

Conclusion
B-727 OFMspert was a very interesting project. It was the first step in generalizing the OFMspert architecture and applying it in an aviation application. Overall the B-727 OFMspert intent inferencing ability matched human interpretations of pilot actions quite well and can be considered a valid model of crew activities for the subset of activities included in the model. The major limitation of the B-727 OFMspert is that it was most effective for modeling discrete pilot activities. Initial analysis of Ames simulator data suggested that those flights with extensive auto pilot use were most suitable and interesting to model. OFMspert, for example, does not represent continuous activities such as 'hand-flying. Post-hoc analysis suggested that OFMspert might better be applied in a 'glass' environment. Thus B-727 OFMspert was a successful first step in pilot activity tracking and set the stage for a follow-on project that specifically addressed OFMspert, automation and mode awareness for the glass cockpit.

The B-727 OFMspert formed part of Serena Connor Verfurth M.S. Thesis. A copy appears in Appendix A. A conference proceeding summarizing the research appears in Appendix B. A journal paper describing this effort is in preparation.

Research on an intent inferencing system for the glass cockpit has progressed significantly over the course of this grant. Advances at both the methodological and software levels have brought this research to near completion. The following will detail progress in both these areas. The methodology is described first, setting the stage for a description of the computer system in its present form. The system, called the Georgia Tech Crew Activity Tracking System (GT-CATS), applies the methodology to understanding the activities performed by pilots as they use automated flight systems to fly a glass cockpit aircraft.

Refinements to the GT-CATS methodology began with narrowing the scope of the inferencing process to focus on the manner in which pilots use automation in the control of the aircraft. Glass cockpit automation uses modes to provide control options to the pilot. Modes enable operators to specify high level constraints on system performance, and, in a given mode, the automation controls low level performance parameters in a specific way. Each automatic mode therefore has characteristics that may lead the operator to select it in a particular situation. These characteristics include the manner in which the mode constrains system performance and the level of operator intervention required to use it.

Advanced cockpit automation incorporates a variety of modes which provide pilots with considerable latitude in managing the flight. However, the proliferation of increasingly complex modes has led to an important problem, namely, pilots can make errors in attempting to use the array of available modes. Mode errors can have serious consequences for flight safety. Indeed, human error in operating complex automated flight modes has played a role in a number of incidents and accidents during the last decade. Nonetheless, modes are useful, and they are likely to proliferate as both the airspace environment and available automation become more complex. GT-CATS research was therefore oriented on the premise that a thorough understanding of mode errors that occur with present cockpit automation is therefore an essential ingredient in understanding how to more effectively develop next-generation automation.

To address this need, the GT-CATS methodology for understanding how pilots use automated modes in the control of glass cockpit aircraft was developed. The capabilities provided by the GT-CATS methodology constitute the set of capabilities required to “track” the activities of pilots. The methodology also supplies a framework for understanding automated modes and their use. Because next-generation automation will undoubtedly also use modes to provide control options to pilots, such a framework is a useful contribution to the development process.

An important purpose of the GT-CATS methodology is to elucidate the factors that give rise to pilot mode errors. Such an understanding is viewed as crucial, particularly if future automation is to effectively incorporate modes that resist and/or tolerate mode errors. The long term aims of
GT-CATS research were therefore identified as helping designers of future automation develop better displays for managing the selection and use of modes, and serving as the source of knowledge for intelligent pilot aiding systems. GT-CATS could also serve as the source of knowledge for intelligent tutoring systems aimed at improving the training received by pilots who use complex flight deck automation.

**The GT-CATS Methodology**

The GT-CATS methodology was designed to track operator activities in real time to produce an 'understanding' of how operators use automated modes to control complex dynamic systems. First, it attempts to predict which level of automation the operator is likely to select, and when, to achieve a desired system state. It also predicts how and when the operator will setup, engage, monitor, and adjust the selected mode. Further, the methodology attempts to understand when the operator is expecting and monitoring an automatic mode transition, and to which mode. Finally, it provides a means for assessing whether the operator is actually performing actions appropriate for the situation.

**OFM/OFMspert.** The GT-CATS methodology is in many ways similar to that embodied in OFMspert, an architecture for a computer-based operator associate in the supervisory control of complex dynamic systems (Rubin et al., 1988). The understanding capabilities (i.e., intent inferencing property) of OFMspert are based the operator function modeling (OFM) methodology. The OFM provides the normative model from which plausible explanations for observed operator actions are generated given the current state of the controlled system. OFMs represent operator control functions using a heterarchical-hierarchical network of nodes. The network heterarchy represents the high-level activities that comprise the operator’s control responsibilities. The hierarchy decomposes activities at the highest (i.e., function) level, into the subfunctions, tasks, and actions required to support the high level control functions (Figure 5). The OFM structure has been successfully demonstrated as effective for describing and prescribing operator activity in complex systems with advanced levels of automation (e.g., Mitchell, 1987).

In OFMspert, knowledge supplied by the OFM is used to construct and maintain a dynamic representation of operator activities. OFMspert uses the blackboard model of problem solving, in which operator functions, subfunctions, and tasks relevant to the current operating situation are posted to a blackboard data structure. Inferencing processes then operate on the blackboard. As actual operator actions are detected, they are understood by connecting them to every posted task they can plausibly support. If the tasks supported by the action support different functions, the action is also assumed to support multiple goals for the time being. OFMspert disambiguates this initial understanding by assessing the action's blackboard connections opportunistically when adequate information becomes available. OFMspert's intent inferencing capabilities have also been validated (Jones et al., 1990).
GT-CATS: Enhancing OFMspert for Systems with Automatic Control Modes. GT-CATS research enhances the OFM/OFMspert methodology to make it suitable for understanding specifically how operators use automatic modes to control complex dynamic systems. As with the OFM/OFMspert methodology, the GT-CATS methodology consists of two components. The first component is an explicit, task-analytic model based on the OFM that normatively specifies how operators use automatic modes to achieve the desired performance from the controlled system. The second component is a mechanism for using the model to understand operator mode selection and operation in real time.

GT-CATS' model is designed to impart an explicit automatic mode orientation to the OFM. Called an OFM for systems with automatic control modes (OFM-ACM), the model specifies normative conditions for undertaking a given activity that reflect standard operating procedures. The model structure, however, is canonical, meaning it describes all the plausible trajectories that may be followed in the course of the operator-automation interaction. This is especially important in systems with automatic control modes, because the operator can select any one of several modes to perform a particular function. When used to predict which mode(s) an operator will select, the normative conditions for selecting a mode are referenced; every possible mode that an operator may select has normative conditions for its use, and every possible mode selection is represented in the OFM-ACM. However, if the operator chooses a mode selection that is not the norm for the current operating situation, GT-CATS uses only the OFM-ACM's canonical structure (i.e., the representation of all feasible choices) to determine that the selected mode, while not the most likely, indeed supports the required function and is therefore valid.

Like the OFM, the OFM-ACM is structured as a heterarchical-hierarchical network of nodes that represent operator activities at each relevant level of abstraction (Figure 6). In the hierarchical dimension, the OFM-ACM decomposes operator functions that must be performed to meet a operational goal into the modes that can be used to perform them, and in turn decomposes each mode into the tasks, subtasks, and actions required to use it depending on the situation. (As with the OFM, such a decomposition is referred to generally as an “activity tree.”) The OFM-ACM's structure is also heterarchical, because multiple functions are typically performed concurrently, and because the use of a particular mode often allows or requires supporting tasks or subtasks to be performed concurrently.

The OFM-ACM also enhances the OFM heterarchy by including an explicit hierarchical decomposition of operator activities for each phase of system control. This enables operator control responsibilities to be represented explicitly in systems whose operation is generally thought of as consisting of several phases, each of which requires operators to address a particular set of control functions. In essence, the OFM-ACM structure includes an OFM activity
tree for each relevant phase of system operation. This enhancement allows the nuances involved with using a given mode to perform a required function in a given phase to be explicated.

GT-CATS uses its OFM-ACM to understand operator mode selection and operation, just as OFMspert relies on its OFM. However, GT-CATS’ mechanism for creating and maintaining a dynamic representation of operator-automation interaction is a departure from the blackboard model of problem solving used in OFMspert. In GT-CATS, the representation is maintained by matching the conditions contained in an instanciation of the OFM-ACM called the Dynamically Updated OFM-ACM (DUO). The conditions in DUO specify when a given activity should normatively be performed in the current operating situation (i.e., the state of the controlled system, the state of the control automation, and high-level goals): A simple top-down, breadth-first search through DUO is used to identify relevant activities. This procedure allows GT-CATS to establish a “best guess” as to what actions will be performed to support which mode selection, tasks, and subtasks. Establishing a best guess enables GT-CATS to immediately disambiguate the reason that an action was performed, and then wait for the action that should be performed next.

A second aspect of GT-CATS’ mechanism for understanding operator mode selection and operation involves comparing actual operator actions to the dynamic representation instanciated in DUO to assess the appropriateness of the actions. This comparison is critical for tracking operator activities, because detected operator actions provide the information required for confirming whether expectations about automatic mode operation have been met, or whether a mode error has occurred. If the operator performs an action that supports a postulated mode, GT-CATS explains the action accordingly. If the operator chooses a mode different from the postulated mode, GT-CATS assesses the detected action to determine if it can be explained as supporting an alternative valid mode. If it can, GT-CATS revises its previously postulated mode selection and begins focusing on the next action that should be performed in support of the mode the operator actually selected.

The GT-CATS Architecture

Once the GT-CATS methodology was firmly established, research efforts focused on the development of a generic architecture that embodies the methodology, and could be used to apply it understanding pilot activities in real time. The generic GT-CATS architecture is depicted in Figure 7. The architecture incorporates the elements necessary to implement the GT-CATS methodology for tracking the activities of operators using automation to control a complex dynamic system. Seven important functional components are shown: the GT-CATS interface, the high level controller (event queue), the state space and high level goal representations, the action handler, the OFM-ACM, and the Dynamically Updated OFM-ACM (DUO) with its associated search procedures.
The arrows in Figure 7 represent information flow between the individual components. The GT-CATS interface receives information about the state of the controlled system and the state of the control automation, as well as operator actions performed on the control automation, from the controlled system. The interface parses this information and sends it to the high level controller for processing. The high level controller schedules appropriate events in its event queue, and initiates processing of these events in accordance with its timing functions. These events include updates of GT-CATS' state space representation, searches of DUO that dynamically update the representation of operator activities, and action handling events. As shown in Figure 3, the action-handler also sends events required to check on actions to the high level controller for scheduling and later processing. The DUO search procedures take information from the state space representation, along with high level goal information, and use it to access the model. The representation built from the model also supplies information to the action handler for use in assessing operator actions. These elements of the generic GT-CATS architecture are now described.

**GT-CATS Interface.** The GT-CATS interface provides a link between the controlled system, its control automation, the control actions undertaken by the operator(s), and GT-CATS proper. Abstractly, it is simply an endless loop that parses the information it receives and sends this information to the high level controller. The information comprises that required to update the state space representation, update the dynamic representation of current operator activities encapsulated in DUO, or signal the action handler to process operator control actions.

**High Level Controller.** The high level controller coordinates the execution of each type of event according to the information sent from the GT-CATS interface. The high level controller includes timing functions that synchronize GT-CATS' to the controlled system in real time, an event queue, and a mechanism for scheduling and processing events in the event queue. As indicated by the arrows in Figure 7, the action handler can also send events to the high level controller for scheduling on the event queue and subsequent processing. On each processing cycle the high level controller processes all the events whose scheduled processing time is at or before the current system time, as dictated by the timing functions.

**State Space Representation.** The state space representation is a dynamic representation of the important state variables that characterize the state of the controlled system and control automation. New state information is received by the GT-CATS interface and scheduled for updating at the appropriate time by the high level controller. On the high level controller's next processing cycle the state space values are updated accordingly to maintain a current representation of the control situation.

**High Level Goals (Performance Specifications).** The representation of high level operator goals for system performance is crucial for determining which activities in DUO are currently
important. The correspondence (or lack thereof) between the high level goals and the current state space is the key indication that a particular set of control activities are relevant in a given situation.

**OFM-ACM, DUO, and Associated Search Procedures.** The OFM-ACM is instantiated in DUO. DUO and its associated search procedures provide the domain and processing knowledge required for understanding operator activities. The high level controller executes an event to update DUO after events that update the current state space have been executed. The search procedures draw information from the high level goals and state space representations to create a summarized description of the current operating situation which is used to reference the conditions in DUO. The search procedure identifies the activities that are relevant to the current operating situation. The set of activities in DUO that are identified as relevant constitute the set of expectations GT-CATS has about the functions, mode selections, and supporting tasks, subtasks, and actions operators should perform to achieve the desired goals. GT-CATS then attempts to confirm these hypotheses using the actual actions performed by operators.

**Action Handler.** The action handler gives and receives information from DUO and the high level controller in order to implement the action handling portion of the GT-CATS methodology. When the GT-CATS interface detects operator control inputs to the control automation, the high level controller schedules the event for processing by the action handler. The action handler also sends the high level controller an event to be scheduled as a result of an action becoming relevant in DUO. Detected actions that cannot be explained on the basis of normative expectations from DUO require the action handler to revise the set of activities that are relevant in DUO. Thus, double-headed arrows link the action handler to both the high level controller and DUO in Figure 3.

GT-CATS' action handler supplies the crucial activity tracking capabilities required understand the actions operators perform when using automatic modes. The action handler's first job is to deal with actions that are deemed relevant for supporting postulated mode selections. The action handler also schedules an event to check that the expected action has indeed been detected after a prescribed time interval. When this event is processed, the action handler notes either that the action was detected during the interval or that the action may be missed or late.

When an expected action corresponds to an actual action performed by a operator, GT-CATS' action handler issues an explanation for the action using DUO. Like the initial expectation of the action, the explanation is a statement to the effect that the action is explained as supporting the subtask, task, mode, and function above it in DUO (i.e., the ones it was expected to support). This explanation indicates that the operator performed the correct action as part of task involved with the normative use of a mode to perform a required function.
GT-CATS' action handler can also produce an accurate explanation for an action that was performed in support of an unexpected, but still valid, mode selection. The canonical OFM-ACM structure (i.e., the representation of all feasible mode selections) instanciated in DUO allows the validity of a selected mode to be confirmed. If the action can be explained accordingly, the action handler explains it, revising its previous expectation in the process. If not, the action handler issues a statement indicating that the action is unexplainable. Conceptually, the explanation revision process works as follows. GT-CATS' action handler first uses DUO to determine if the action does support a mode that could be used to perform the required operational function. It then determines if the mode is in fact in use. If the task and subtask that the action supports, according to DUO, are now irrelevant (i.e., the operator(s) are not addressing them, because they have already been accomplished), then the action is explained as supporting the operator's mode selection option.

**Applying GT-CATS to the Glass Cockpit**

Glass cockpit aircraft are characterized by advanced flight control automation that offers multiple modes for controlling the flight path of the aircraft at different levels of automation. In developing GT-CATS, some important assumptions were made about the glass cockpit domain. First, GT-CATS assumes that Air Traffic Control (ATC) clearances are received via datalink technology. Datalink effectively automates the process by which clearances are acknowledged by the pilots. Other assumptions address the contents of the OFM-ACM. These assumptions direct the focus of the OFM-ACM on the use of automation modes for controlling the aircraft. Finally, it is assumed that data about pilot control actions, important state variables, and information programmed in the automation is available, and that the automation always works as advertised. Given these assumptions, the generic GT-CATS architecture depicted in Figure 3 above was implemented for the glass cockpit. The GT-CATS architecture as implemented for aviation is shown in Figure 4 below. The GT-CATS interface and high level controller work the same way as described above for the generic GT-CATS architecture. The other elements of GT-CATS for the glass cockpit have been tailored to suit the domain.

In the glass cockpit, high level goals of the flight are dictated by the desired flight path. Because in many cases the specific flight path chosen is left to the discretion of the pilots, the desired flight path is really an “envelope” defined by ATC clearances and the performance limitations of the aircraft. In GT-CATS, the desired flight path is therefore represented by a data structure called the “limiting operating envelope.” The limiting operating envelope summarizes the preplanned flight path along with dynamic deviations specified by ATC. Together, this information comprises that which is required to dynamically represent the high level goals of the pilots.
The GT-CATS state space consists not only of aircraft state variables (e.g., heading, altitude, airspeed, etc.), but also of autoflight system state variables (e.g., target values, engaged modes) and programmed information. Information about flight progress (e.g., top-of-climb point passed or not) is also included. This additional state information is essential for generating hypotheses about which modes pilots should use to change the flight path of the aircraft when such changes are required.

As in the generic architecture, the OFM-ACM is first instanciated in DUO. DUO and its associated search procedures then provide GT-CATS with knowledge required for understanding pilot activities. The search procedures draw information from the limiting operating envelope and state space representations to create a summarized description of the current operating situation which is used to reference the conditions in DUO. The search procedure identifies the activities that are relevant to the current operating situation. This set of activities in DUO constitute the set of expectations GT-CATS has about the functions, mode selections, and supporting tasks, subtasks, and actions pilots should perform to achieve the desired flight path, as specified by the limiting operating envelope. GT-CATS then attempts to confirm these hypotheses using the actual pilot control actions performed on the autoflight system of the aircraft.

GT-CATS' action handler is responsible for attempting to understand the actions pilots perform when using complex autoflight system modes. When an action is identified as relevant in DUO, the action handler first schedules an event to check that the pilot has actually performed the action within a prescribed time interval. If the pilot fails to perform the action by the time the event is processed, the action handler notes that the action may be missed or late. If the pilot performs the action expected according to DUO within the prescribed time interval, the action handler issues an explanation for the action. The last function of the action handler is to produce an explanation for a pilot action that supports an unexpected, but still valid, mode selection. For this purpose, DUO instanciates the canonical structure of the OFM-ACM (i.e., the representation of all feasible mode selections). GT-CATS' action handler then refers to DUO to process the unexpected action. If the action supports a valid mode selection, the action handler issues an explanation for it, revising its previous expectations in the process. If the action is not valid, the action handler issues a statement indicating that the action cannot be explained.

Summary

GT-CATS works by using its OFM-ACM to determine what functions pilots should perform to acquire and track the desired flight path and hypothesize which automated mode(s) will be used to perform these functions. It also hypothesizes the tasks, subtasks, and actions pilots will perform to properly use the postulated mode(s). Given these hypotheses, GT-CATS attempts to explain pilot actions. Four additional conceptual elements augment these basic elements of the methodology in the GT-CATS glass cockpit implementation, viz., a means of dynamically representing the
desired flight path (high level goals), a means of maintaining an updated state space representation in real-time, and a high level controller to perform real-time event processing.

Validation

GT-CATS research has also addressed the development of a procedure for validation of GT-CATS. The GT-CATS methodology will be validated through a series of empirical studies designed to demonstrate its use and effectiveness in understanding how pilots use complex autoflight system modes. These studies will follow the general procedure set forth by Jones et al. (1990). The validation will proceed in three phases. First, pilots acquainted with the autoflight system of the glass cockpit will serve as subjects for pilot studies in which the adequacy of the part task simulator and experimental scenarios will be assessed. After the pilot studies are completed, NASA Ames Advanced Concepts Flight Simulator data will be analyzed by GT-CATS to determine if GT-CATS can interpret the actions performed by test pilots during the NASA flight scenarios. The final phase of validation will involve using pilots as subjects flying a Boeing 757 autoflight system part-task simulator linked to GT-CATS in real time. GT-CATS' explanations for pilot actions in each experimental scenario will be compared to the pilots' own explanations to determine the degree to which GT-CATS' explanations match those of the pilots. Pilots will also be asked to provide verbal protocols of explanations for their actions which will be compared to the explanations from GT-CATS. GT-CATS' explanations should correspond closely to the pilots' explanations, if GT-CATS effectiveness is to be confirmed.

The VNAV Tutor (Ed Crowther and Alan Chappell)

One of the major tasks of pilots of modern aircraft is monitoring and understanding the status and behavior of the auto flight system, i.e., mode awareness. Control modes of the system change due to pilot commands (manually) or in response to system events (automatically). The flexible and dynamic nature of the system increases both the functionality of the control system and the cognitive demands placed on the pilot. In order to maintain mode awareness in this dynamic environment, the pilot must be continuously vigilant of indications from several locations within the cockpit. Lacking accurate and complete system knowledge and/or an interface that clearly presents the system state and constraints, the pilot may misunderstand the control modes. Pilots often cite vertical path navigation (VNAV) as a flight management system function that surprises them. The VNAV Tutor, a computer-based training system, was developed to address this issue. The VNAV Tutor attempts to improve the pilot's understanding of VNAV control modes and the interaction of the mode control panel functions with the flight management system during VNAV usage. Furthermore, it attempts to help pilots build a robust conceptual model of vertical navigation operation (Figure 8). An evaluation showed that the VNAV Tutor enhanced both the conceptual understanding and use of the vertical navigation function by pilots transitioning to
aircraft with sophisticated auto flight systems. Appendix C contains a copy a manuscript submitted for publication that describes the VNAV Tutor and associated evaluation.

3D Primary Flight Display with Terrain Information

An important world-wide aviation safety problem is still the controlled-flight-into-terrain or CFIT accident. Area navigation and onboard terrain elevation data bases offer the potential for improved cockpit displays near by terrain.

This project has developed a prototype primary flight display format designed to re-enforce the pilot's model of both lateral and vertical navigation in near-terrain situations. This new display format is referred to as the Spatial Situation Indicator (SSI). Specific emphasis has been placed on the terminal phase of flight with terrain modeling in the vicinity of the departing and destination airport.

The unique design incorporated perspective symbology which depicts a prediction of the aircraft's predicted position and terrain clearance information for up to 75 seconds ahead of the aircraft. Projection of the flight path is based on a “fast time” modeling technique described by Grunwald (1985). Traditional flight paths use the “tunnel in the sky" approach which present no reference to the ground elevation e.g., Grunwald (1982). The technique developed for this research utilized roll stabilized vertical lines "whiskers" positioned at 15 second intervals out to 75 seconds. Figure 9 illustrates the virtual “whiskers” and flight path. The whiskers are displayed in pairs of equal distant widths so that in steady level flight a perspective path is projected. The whiskers are color coded using green and yellow. The green lower portion extends from the predicted aircraft altitude at that interval to the terrain below. Its length therefore is a direct representation of the terrain clearance at that point in the aircraft’s path given no changes in aircraft flight path.

The display also incorporates a dynamically color coded terrain grid. The color coding is based upon aircraft predicted height and terrain spot elevations. The color coding uses dark green for safe terrain and dark red for dangerous terrain. The terrain grid is comprised of a triangular mesh with each triangle having sides of 2 nautical miles (NM). Man-made obstructions such as radio towers are also shown on the terrain grid. Information for building the terrain and obstruction files is obtained from the approach plates for each runway in the scenario.

An experimental evaluation of the display was conducted on-site at a major U.S. airline. Experimental participants are current glass cockpit flight instructors. Each experimental subject, after training to familiarize him/herself with the part-task aircraft simulator and interface, will fly three scenarios based on actual controlled flight into or toward terrain as described by Bateman (1991).

Each experimental participant used one of the three displays: the baseline cockpit display, the primary flight display, and this display with flight path predictor and ground terrain
information. A total of eighteen pilots will participate, six with each display. Attention diverting tasks are implemented to match as closely as possible the scenarios as they are described by Bateman. ATC communications are implemented using simple voice communications without supporting electronic intercoms. The experimenter carries out the air traffic controller (ATC) communications. The goal of the experiments is to measure how quickly pilots can detect dangerous terrain with the three different display formats. Response time of the pilot for corrective action is recorded as well as MCP inputs. Data from this experiment are still being analyzed. The project formed a portion of Jim Williams’ M.S. thesis which is in preparation.

Conclusion
This research encompassed numerous projects, some of which are still on-going; it also provided the foundation for new efforts. GT-CATS is being completed under a follow-on grant with NASA Ames. In addition, the VNAV Tutor provided essential insight and experience for exploring training issues and the role of computer-based tutors in aviation. On-going work greatly benefits from the VNAV Tutor development and evaluation.

This grant allowed the various members of the Center for Human-Machine Systems Research Center group at Georgia Tech to begin to become 'aviation-literate'. Although challenging at times, it has been intellectually stimulating. To some extent, the enrollment of two Delta pilots in our doctoral program attests to the quality of our initial work. We hope to contribute significantly to the basic knowledge on human-centered aviation on the flight deck.

Finally, in Appendix D, is enclosed a copy of a chapter to appear in Bill Rouse’s series on Human Interaction with Complex Systems. Though not aviation-related, this chapter describes the MSOCC project in which both the operator function model and OFMspert were initially developed. Various pieces of the MSOCC project were undertaken with the financial support of NASA Ames under a previous grant and the continued intellectual stimulation of Ev Palmer

References

Appendices
(Contact Principal Investigators for Copies of the Appendices)

Appendix A


Appendix B


Appendix C


Appendix D

Figure 5. Generic OFM structure
Figure 6. Generic OFM-ACM structure showing explicit phase/subphase activity decomposition and mode selection level
Figure 7. Generic GT-CATS Architecture
Figure 8. Vertical Profile Display: The Heart of the VNAV Tutor
Figure 9. The SSI Display