

THE CREW ACTIVITY TRACKING SYSTEM: LEVERAGING FLIGHT DATA FOR AIDING, TRAINING AND ANALYSIS

Todd J. Callantine, San Jose State University/NASA Ames Research Center, Moffett Field, CA

Introduction

Aviation safety is currently receiving unprecedented attention. Spurred by projected growth in air traffic, practitioners and researchers alike are seeking ways to ensure that aviation systems continue to operate safely. Studies implicate human error as the primary cause in 70% of hull-loss accidents worldwide [1]; more flights create more opportunities for disaster. New operational concepts designed to meet increased demand necessarily place greater reliance on automation to improve efficiency (e.g., [2]). However, in addition to requiring changes in operator roles and responsibilities that require careful scrutiny, automation can increase cognitive workload and foster errors that can lead to unsafe operations (e.g., [3-5]).

To better understand human error and its context while addressing safety challenges due to increasing traffic density, the aviation research community needs data. Much has been learned from pilot narratives submitted to the U.S. National Aeronautics and Space Administration (NASA)-administered Aviation Safety Reporting System (ASRS) (e.g., [6]). However, detailed error data are rare; test subjects in high fidelity simulations may not commit consequential errors (e.g., [7, 8]). Unfortunately, catastrophes therefore remain an important source of information about errors. Airlines are also seeking ways to address safety concerns, including errors. The U.S. Federal Aviation Administration (FAA) Advanced Qualification Program (AQP) seeks to qualify, train, and certify flight crew members and operational personnel to a high level of proficiency [9]. As with research efforts, AQP could benefit from data about real errors.

Technological advances have made flight data a viable real-world data source for studies of human error and error prevention; hundreds of parameters are currently available for analysis. These data have enabled airlines to institute increasingly advanced Flight Operational Quality Assurance (FOQA) programs, which analyze flight data from line operations to detect “operational irregularities that can foreshadow accidents and incidents [10, p 22],” and proactively disseminate this information to flight crews and maintenance personnel. Central to FOQA are systems for analyzing operational flight data and maintaining databases of trends. Many commercially available Ground Data Replay and Analysis Systems (GDRASs) now include three-dimensional animations and other sophisticated tools [11]. Efforts to further enhance and standardize event detection and other analysis capabilities are ongoing in the FAA’s Aviation Performance Measuring System (APMS) program [10, 12, 13].

This paper presents an intent inference technology, referred to as *activity tracking*, that in the future could also support flight-data-driven safety-enhancement efforts. A methodology for activity tracking has been implemented and validated in the Crew Activity Tracking System (CATS) [14, 15]. As implemented for the flight deck, CATS uses knowledge about the pilot’s task and the current operational context to predict nominal activities and interpret actual pilot actions. By analyzing pilot action data in conjunction with clearance constraints and other flight data parameters, CATS can help disambiguate errors from other causes of abnormal flight conditions, and characterize error-inducing contexts in operational terms.

Furthermore, because CATS interprets actual pilot actions *vis a vis* a model of nominally correct operations, CATS can track when pilots deviate

from 'preferred' procedures represented in the model. Such deviations may not lead to operational irregularities *per se*, but they may offer insights about how pilots understand the operation of their aircraft and, perhaps, how airlines might realize benefits from modifications to current error-inducing aircraft systems and interfaces. In addition, aviation research may at times hinge upon small issues that activity tracking could potentially elucidate.

CATS' output was originally designed to directly support training and real-time aiding applications [14]; as part of a computer-based training system, for example, CATS can detect errors and suggest the nominal correct action to the student. As the student improves, the training system can evolve into an aid [16], scaffolding skill development. Using CATS as part of FOQA-supported AQP, managers could 'automatically' introduce the precise context in which errors identified by CATS occurred, into a training curriculum. Increasing the efficiency and fidelity of information transfer to pilots in this way can potentially yield safety benefits.

The paper first reviews the current state-of-the-art in FOQA data collection and analysis. It then discusses activity tracking, and introduces CATS. After describing CATS' knowledge representations and activity tracking method, the paper describes using CATS for analysis of flight data, and for training/aiding. The paper concludes with a discussion of issues surrounding activity tracking with flight data and current research efforts to demonstrate CATS using flight data from the NASA Langley Boeing 757 (B757) Airborne Research Integrated Experiment System (ARIES) aircraft within the FAA/NASA Aviation Safety Program.

FOQA Programs

Airlines institute FOQA programs to improve safety by exploiting the wealth of flight data they create during line operations. The idea has been evolving since Flight Data Recorder (FDRs) became mandatory equipment on aircraft; aircraft manufacturers currently deliver aircraft equipped with FOQA flight data collection equipment. The objective of a FOQA program is detect—"early enough"—"technical flaws, unsafe practices or

conditions outside of desired operating procedures [so-called deviations, or 'exceedances'] to allow intervention to avert accidents or incidents [10, p. 3]."

Airlines outside the U.S. have been pursuing FOQA-type programs for over three decades. Since 1995, the FAA has been promoting the voluntary implementation of FOQA by U.S. Airlines. Estimates place safety- and maintenance-related benefits at over a million dollars annually for large fleet sizes, excluding costs that would be incurred due to the stigma that would likely result from a serious incident or accident. To date, nine major U.S. carriers have implemented FOQA, to the extent that aircraft in their fleets are equipped with Quick Access Recorders (QARs) to allow efficient transfer of flight data to GDRASs [10].

Successful FOQA depends on the effectiveness with which deviations can be detected and communicated to flight crews and other operational personnel. GDRASs therefore play a crucial role. In addition to data integrity filters, configurable event specifications, parameter graphs, and instrument visualizations, newer GDRASs incorporate databases of navigation information, including charts with current position overlays, three-dimensional scenes, and database query functionality. Some also include synchronized digital video/audio and VCR-like controls, so that training simulation data may be analyzed in detail using the same tool. This allows direct comparison of deviations observed during training with those detected from operational flight data.

The FAA's Aviation Performance Measuring System (APMS) represents an attempt to further enhance GDRAS capabilities by enabling examination of not just deviations, but all flight data. APMS seeks to objectively and continuously evaluate a flight crew's technical performance, and automatically convert these data into useful safety information via data mining and other knowledge-based techniques [10, 12, 13]; however, technical information on APMS is scant. Standardizing event detection capabilities could offer benefits as it could enable FOQA information to be shared among airlines.

Current FOQA programs differ across airlines. Differences include the number and types of flight data parameters collected, the types of deviations

the data are inspected for, and the manner which trend information is cataloged and used. Tables 1 and 2 provide a flavor for the range of parameters and categories of deviations used in basic FOQA programs, and suggested for “state-of-the-art” programs. Table 1 lists some flight data parameters deemed useful for FOQA in the descent/approach phase of flight [17] (see [17] for complete lists for all flight phases). For advanced FOQA, Table 1 reflects an increased emphasis on the aircraft’s automated systems, in keeping with the increasing recognition of automation’s error-inducing potential. Information about the Flight Management Computer (FMC)—including the appearance of each pilot’s Control and Display Unit (CDU)—is especially useful, as Flight Management Systems (FMSs) play an ever-larger role in flight operations.

Table 2 lists an assorted sample of event (deviation) categories that FOQA programs search for in descent/approach data (again, see [17] for a comprehensive list). Table 2 reveals an expansion of deviation categories to include safety-critical

systems, such as the Ground Proximity Warning System (GPWS). Although Table 1 suggests collecting FMC/CDU data, exactly how these data will be used is apparently unclear, as the expanded set of deviation categories in Table 2 does not include any categories that directly relate to these data.

As implemented by a major European carrier [18], FOQA is viewed as an objective data source that complements several other subjective data sources (e.g., pilot reports). But FOQA cannot detect some types of deviations (e.g., runway incursions), and alone cannot attribute cause to a deviation. All data sources together do not achieve 100% risk coverage. However, FOQA has proven effective in detecting and remediating several

Table 1. Sample Descent/Approach Parameters (from [17])

Used in Current Programs	Additionally Suggested for State-of-the-Art Programs
Altitude	Autopilot Flight Control System mode annunciation
Barometric altitude	Autopilot engage status
Computed airspeed	Autothrottle engage status
Flap position	EFIS format/display
Gear in transit	EICAS format/display
Mach speed	FMC winds
Normal acceleration	FMC display (both pilots)
Pitch attitude	Microphone keying
Relative time	NAV receiver frequency
Roll attitude	Pilot event marker
Vertical speed	TCAS event

Table 2. Sample Descent/Approach Event Categories (from [17])

Used in Current Programs	Additionally Suggested for State-of-the-Art Programs
Approach speed High/Low	Abnormal flight control position
Approach thrust	Auto-brake status
Bank angle limit	Go around
Flap placard speed	Ground Proximity Warning System
Glideslope deviation High/Low	Ground spoiler not armed
Gross power increase on final	Lateral acceleration
High descent rate	Pilot event marker
Localizer deviation	Reduced lift margin
Low power on final	Speed below 10,000 ft
Reversers deployed in flight	Stall avoidance
Speed deviation at threshold	Stick shaker
Speedbrake arm delay	TCAS
Wind shear below 1500 ft.	Vertical acceleration

classes of deviations, including long landings, non-stabilized approaches, asymmetrical thrust settings at takeoff, and pilot-induced oscillations.

Activity Tracking

Activity tracking differs from the detection and analysis of deviations. The activity tracking methodology involves first predicting the set of expected nominal operator activities for the current operational context, then comparing actual operator actions to these expectations to ensure operators performed correct activities. In some situations, various methods or techniques may be acceptable; therefore the methodology also includes a mechanism for determining that, although operator actions do not match expectations exactly, the actions are nonetheless correct.

In this sense, CATS is designed to 'track' flight crew activities in real time and 'understand' that they are error-free. Such capabilities are necessary for the development of 'intelligent' training systems, which could lessen the need for pilots to learn important skills 'on the line.' They are also required for 'intelligent' aiding systems, which could provide crews with timely advice and reminders, and help eliminate tragic pilot-induced accidents. Finally, these capabilities might enable airlines to perform offline, crew-centered analyses of FOQA data in greater detail than is possible with current GDRASS.

However, in addition to parameters that define the state of the controlled system (as with FOQA parameters), activity tracking also requires data about the dynamic set of constraints on controlled system behavior, as well as data about actual operator actions. While constraint data in the form of data linked clearance information will likely be widely available in the near future, a number of legal issues still impede the release of pilot action data [10]. This paper takes the view that the promise of significant safety benefits can help overcome these issues in the future. Activity tracking also requires a valid model of nominally correct operator activities suitable for deriving the set of 'preferred' operator actions predicted (expected according to the nominal model) in the current operational context. For the flight deck, such models may be adapted from extant AQP models and validated in high fidelity simulations.

Crew Activity Tracking System (CATS)

CATS implements a methodology for activity tracking in a computer-based system that has been validated to work in real time [14]. Figure 1 depicts the CATS architecture and processing method generically. As described above, CATS uses representations of the current state of the controlled system and constraints on its operation to derive the current operational context. CATS then uses this representation to generate predictions from the activity model (①). CATS compares detected operator actions to its predicted activities (②), and it assesses actions that it cannot directly interpret using the predictions by periodically referencing the activity model until enough new data has arrived to disambiguate possible interpretations (②). Thus,

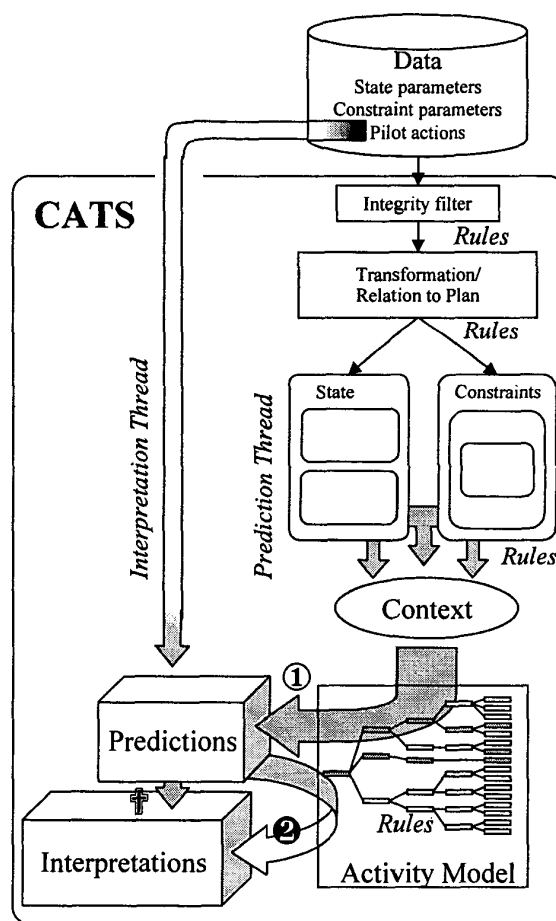


Figure 1. Generic CATS Architecture And Processing Method.

two threads comprise the activity tracking methodology as implemented in CATS: a 'prediction thread' responsible for generating the context information necessary to predict nominal activities, and an 'interpretation thread' that interprets operator actions.

Figure 1 differs from previous depictions (e.g., [14, 19]) in that it places special emphasis on the process CATS uses to 'condition' actual flight data. State and constraint data are first filtered for integrity, i.e., filters process the stream of high-frequency parameter values that comprise the data and remove any inconsistencies or invalid values. This step is not necessary for flight data that have already been verified. The next step is to create generic references to specific data values (e.g., 'next waypoint,' as opposed to 'waypoint ABC'). These 'relative' identifiers are necessary for constructing a representation of the current operational context that can be used to reference a nominal model without loss of specificity.

The next sections describe how an implementation of CATS for the flight deck works. The first section presents a simple example scenario. Subsequent sections detail the knowledge representations CATS requires, and the process CATS performs to track some flight deck automation configuration and usage activities pilots could perform in this situation.

These sections all assume some knowledge of commercial aviation and a Boeing 757-style autoflight system. The basic scheme is that pilots first program the flight plan into the FMS via the CDU. After engaging the autopilot and autothrottles, they interact with aircraft's Mode Control Panel (MCP), setting limits/tactical targets and engaging pitch, roll, and thrust modes as required to comply with air traffic control clearances. High-level modes such as Lateral Navigation (LNAV) and Vertical Navigation (VNAV) track the FMS-programmed plan; other modes, such Flight Level Change (FLCH), achieve a tactical target state (the MCP target altitude, in the case of FLCH). A detailed description of the Boeing 757 autoflight system mode usage is provided in [14]; see [3-5] for discussions of mode errors and automation issues.

Example Scenario

Figure 2 shows a simple flight scenario. An aircraft is descending along an FMS-computed vertical path using a path-following sub-mode of VNAV, called VNAV PTH. The path includes a crossing restriction at a waypoint ('XYZ'). The aircraft has reached an altitude of approximately 8000 feet when the crew receives the clearance "expedite descent to 5000 feet; comply with restrictions." This clearance indicates that the crew must now descend to 5000 feet as rapidly as possible, yet still slow to 200 knots indicated airspeed at XYZ. The crew must take action, as the air traffic controller has clearly found the rate of descent produced by VNAV PTH insufficient for the current traffic situation.

This scenario, while extremely simple, is an example of one in which pilots can exhibit misconceptions about the performance of the VNAV mode [20]. One action flight crews may perform is to "extend airbrakes... expecting an increase in the rate of descent," when in fact, this action "results in an increase in thrust to maintain the selected speed in the presence of additional drag [20, p. 15]"—an undesirable outcome.

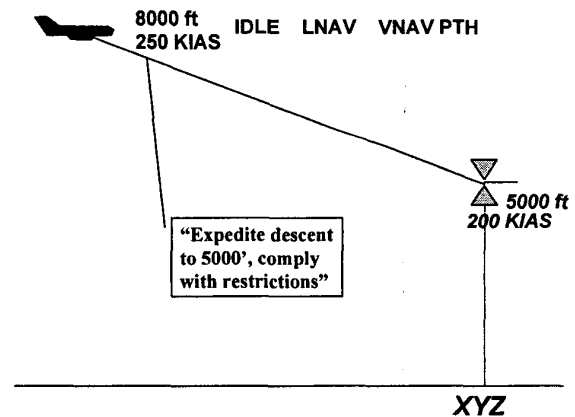


Figure 2. Example Scenario.

State Space

CATS represents states as hierarchical whole-part relationships. For example, it represents the state of the FMS in part as a sequence of waypoints, each with its attendant latitude, longitude, and speed and/or altitude restrictions (e.g., '210 knots at

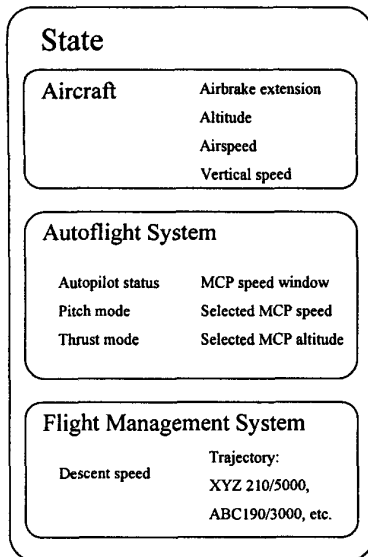


Figure 3. State Space For The Example Scenario.

or above 7000 feet’). Figure 3 shows some state parameters pertinent to the example.

Constraints

Constraints bind the trajectory of the aircraft, and include Air Traffic Control (ATC) clearances. Although clearances typically specify states to be achieved, or trajectories to ‘trace,’ tolerances exist around the objective state, so that the objectives specified by clearances actually form part of a so-called ‘limiting operating envelope.’ For example, a clearance to ‘expedite descent to 5000 feet’ in fact places a set of constraints on the next segment of the flight path. The constraints are those implied by the greatest rate of descent possible for passenger comfort, plus the requirement that the aircraft is within, say, 250 feet of the altitude of 5000 feet when level flight is reestablished. Like states, CATS represents constraints as hierarchical whole-part relationships.

Constraints on the aircraft’s trajectory are an important addition to the typical FOQA data that CATS requires. With the advent of data link technology, such information will increasingly be available digitally. Future data link message sets might include specifics about the constraints that a particular clearance affects, and extant constraints

that a new clearance overrides; these data would

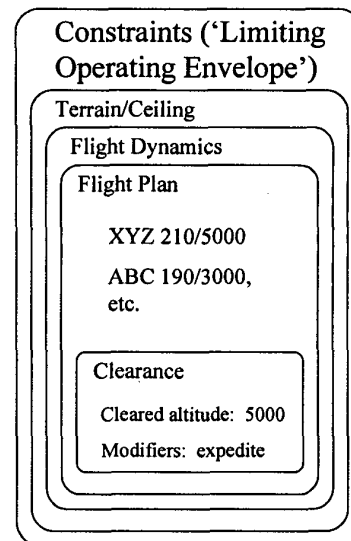


Figure 4. Constraints For The Example Scenario.

then become part of the complete FOQA data set. Figure 4 illustrates constraints in effect immediately after the crew receives the ‘expedite’ clearance.

Context

For purposes of activity tracking, operational context is an operationally relevant collection of state parameters and operational constraints in the current situation, variables derived from these data, and a collection of relationships between actual and/or derived states and constraints. CATS summarizes context knowledge crucial to its activity tracking application using Boolean-valued ‘context specifiers.’ A context specifier is simply a clause that describes the current value of a state or constraint, or the relationship between state(s) and constraint(s). Thus, because the states and constraints form hierarchies, the context specifiers for a particular operational setting also form a hierarchy. The context specifier ‘FMS-trajectory-within-limits,’ which specifies that the current FMS-programmed trajectory matches that which would be required to use the FMS to meet the current flight trajectory constraints, subsumes the context specifier ‘FMS-descent-speed-within-limits,’ because the descent speed may be considered part of the FMS trajectory.

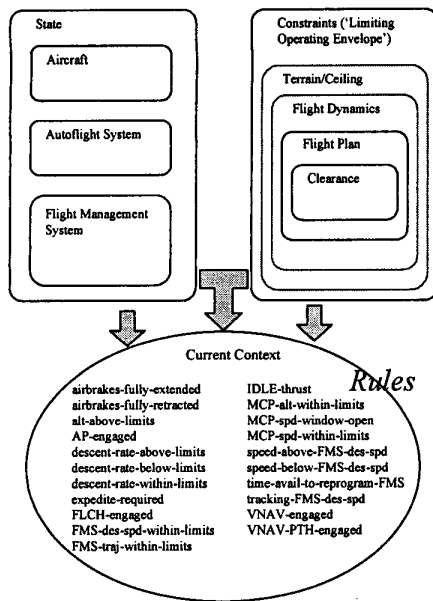


Figure 5. Context Generation, And 'Context Specifiers' Pertinent To The Example Scenario.

CATS can efficiently summarize the current operational context by assessing the lowest level context specifiers, and assigning the parents of those that evaluate true a value of true as well. Each low-level context specifier has rules that express when it is true for the current operating context, which then determines when its parents are true. Figure 5 depicts the information CATS uses to generate context specifiers, and the list of context specifiers pertinent to the example scenario (note that some of these context specifiers are mutually exclusive, and cannot all evaluate true at once).

Some context specifiers present a challenge to evaluate accurately. For example, the context specifier 'time-avail-to-reprogram-FMS' is included because it influences a flight crew's decision to opt for a lower-level, tactical autoflight mode instead of continuing in VNAV, for example. Individual pilots likely evaluate their 'mental equivalent' of such a context specifier differently, given many additional contextual elements. These elements include the nature of other activities that they are currently performing, or need to perform, and the perceived proficiency of themselves or the other crew member at making FMS entries. It also

requires accurate predictions about the future state of the aircraft. The closing discussion addresses other methods for evaluating context specifiers of this sort, beyond the simple heuristics employed currently.

Activity Model

CATS uses a computational model of operator activities that represents both preferred and correct alternative methods for accomplishing system objectives. The CATS model is a normative model based on the Operator Function Model (OFM) (see [14, 15]), that allows high-level activities to be decomposed as necessary to adequately represent the human-machine interactions of interest, down to the level of specific operator actions. Each activity is represented as containing conditions (rules) under which operators should nominally perform it. The conditions take the form of AND/OR trees comprised of context specifiers. Thus, as depicted in Figure 1, the CATS knowledge representations effectively apply several layers of rules. This helps ease the modeling process, because a context specifier, once defined, serves as a 'macro' that may be used in the conditions for any activity, just as a derived variable may be used in the rules for evaluating various context specifiers.

Figure 6 depicts a CATS model fragment that can be applied to the example scenario; Table 3 provides a list of the conditions for predicting the numbered activities under nominal conditions. The activities whose conditions evaluate true given the current (most recently updated) set of context specifiers are those that the model nominally predicts the crew will perform in the current context. Because the search for predicted activities proceeds top-down, CATS predicts low-level activities only if it has already predicted their parent activities. This means that high-level conditions need not be repeated in lower-level activities to predict them (although this practice sometimes aids clarity). This 'memoryless' feature of the model—that the model can produce the nominal set of required activities for given a contextual snapshot—makes it a powerful tool for supporting 'what-if' queries. By simply adjusting the context (or further 'upstream' states and constraints), an adjusted set of nominally predicted activities is produced. And while the model is normative, including accurate temporal context specifiers potentially makes it

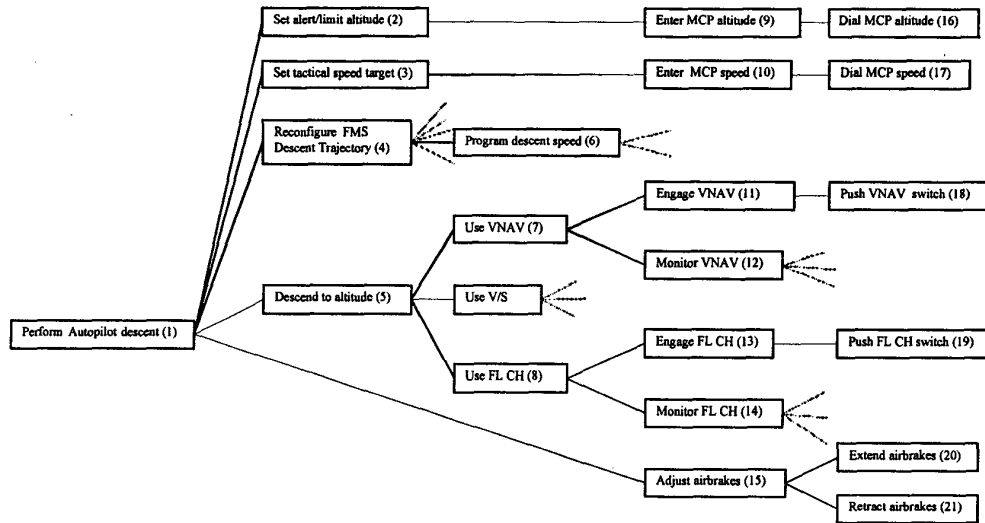


Figure 6. CATS Model Fragment For The Example Scenario.

Table 3. Conditions For Nominally Predicting Activities In The Model Fragment Shown In Figure 6.

- (1) (AND AP-engaged alt-above-limits)
- (2) (NOT MCP-alt-within-limits)
- (3) (AND MCP-spd-window-open (NOT MCP-spd-within-limits))
- (4) (AND (NOT FMS-traj-within-limits) time-avail-to-reprogram-FMS)
- (5) alt-above-limits
- (6) (NOT FMS-des-spd-within-limits)
- (7) (AND (NOT expedite-required) (OR FMS-traj-within-limits time-avail-to-reprogram-FMS))
- (8) (OR expedite-required (AND (NOT FMS-traj-within-limits (NOT time-avail-to-reprogram-FMS))))
- (9) (NOT MCP-alt-within-limits)
- (10) (NOT MCP-spd-within-limits)
- (11) (AND (NOT VNAV-engaged) MCP-alt-within-limits)
- (12) VNAV-engaged
- (13) (AND (NOT FLCH-engaged) MCP-alt-within-limits)
- (14) FLCH-engaged
- (15) (OR (AND (NOT descent-rate-within-limits) (NOT VNAV-PTH-engaged)) (AND (NOT tracking-FMS-des-spd) VNAV-PTH-engaged))
- (16) (NOT MCP-alt-within-limits)
- (17) (NOT MCP-spd-within-limits)
- (18) (NOT VNAV-engaged)
- (19) (NOT FLCH-engaged)
- (20) (AND (NOT airbrakes-fully-extended) (OR (AND IDLE-thrust VNAV-PTH-engaged speed-above-FMS-des-spd) (AND (NOT VNAV-PTH-engaged) descent-rate-below-limits))))
- (21) (AND (NOT airbrakes-fully-retracted) (OR (AND IDLE-thrust VNAV-PTH-engaged speed-below-FMS-des-spd) (AND (NOT VNAV-PTH-engaged) descent-rate-above-limits))))

more responsive to contextual subtleties (cf. [21]). The model also usually includes information about which pilot should nominally perform an activity (unless a separate model is used to track each pilot's activities).

Specifically, the model fragment in Figure 6 represents the activities of setting target speeds and altitudes using the MCP, reprogramming the FMS to reflect the current flight plan constraints, engaging one of three possible, mutually exclusive vertical modes (VNAV, FLCH, and Vertical Speed

(V/S)), and adjusting the airbrakes. For the example scenario depicted Figure 2, and the accompanying CATS knowledge representations, the model would predict that the pilot responsible for making automation inputs should push the MCP FLCH switch (as the required altitude 5000 feet is presumably already set). As FLCH engages, and the requested rapid descent commences, the set of context specifiers will reflect this change, and predict that the pilot should monitor the FLCH descent. The next section focuses on what happens in the example when the crew performs other activities instead.

Action Interpretation

As shown in Figure 1, two threads comprise activity tracking. First, the prediction thread predicts activities an operator is likely to perform given the current operational context, as described in the previous section. Then, after the operator actually performs some action, the interpretation thread processes the action to determine whether it supports predicted activities, or some acceptable alternative. An operator error may be signaled if an action does not support any acceptable methods for meeting current operational constraints, or if no action occurs to support a needed activity within some specified interval. If CATS cannot interpret an action immediately, it will try again periodically as it receives new data.

The example scenario is marked by the possibility that flight crews will attempt to increase the rate of descent using the airbrakes, when in fact this action only slows the aircraft along the path, and causes the throttles to advance, if necessary. Because the airbrake extension action does not support any activities that CATS initially predicts according to the model, the interpretation thread must determine if the action is viable for any other reason in the current context. In the example, the interpretation thread checks conditions for using the airbrakes, and finds no correspondence with the current context, signaling a possible error.

Suppose, on the other hand the flight crew opts to engage V/S mode (required actions and conditions omitted from Figure 6 and Table 3). As long as the crew sets an adequate descent rate in the MCP vertical speed window, this course of action is acceptable—although using FL CH is preferred according to the model. Thus, CATS can be used to

capture the context when crews choose non-preferred actions, so safety managers can assess action preferences under specific conditions in detail.

Applications

Analysis

While CATS was not initially developed as an analysis tool, in a research environment, this application has proven the most beneficial. New flight deck procedures designed for use with new operational concepts have been analyzed using CATS with high-fidelity full-mission simulation data [19, 22, and 23]. For these applications, CATS has been augmented with visualization interfaces designed to provide insights into crew performance, much like FOQA GDRASSs.

In particular, CATS has proven useful for analyzing crew performance in the NASA Ames Advanced Concepts Flight Simulator (ACFS), a full-motion, visual-equipped flight simulator. The ACFS sends CATS detailed data in real time, so CATS can track crew activities as they are performed. When CATS detects that a crew is experiencing difficulties implementing a new procedure, it can also help determine why. Because predictions are made as soon as the operational context allows the activities to be performed, activity latencies can be determined by comparing the time when the actual activities are performed against the time when they are predicted. Activities that are not performed soon after the predicted time may be inadequately cued. Other activities may confound performance, suggesting conflicting task and resource demands. Its accompanying visualization interfaces and data-replay capabilities make CATS a useful tool for debriefing subject pilots.

Training/Aiding

CATS also supports the applications that motivated it originally, acting as the source of knowledge for context-sensitive training systems and operator aids [24]. With its model of operator activities, CATS can provide instructions to the operator phrased from the viewpoint of the operator. The prototype aiding system described in [24] suggests that this a powerful advantage; a few simple lines of text are all that is required to cue the flight crew to the next required activity, and support

queries about why and how the crew should perform the activity. At the research level, this scheme is useful for eliciting feedback from operators about how the underlying model and sets of rules for encapsulating context information should be structured.

Discussion

This paper has focused on the potential for using CATS as a FOQA analysis tool, and provided an example of how CATS works to uncover subtle, yet important, distinctions between different methods of operation in a particular context using a model of nominal activities. While GDRASs that detect deviations could doubtless also be tailored to detect the ‘incorrect use of airbrakes’ issue discussed above, given information about the ‘expedite’ clearance, a CATS-based tool could potentially *reveal* such issues, without knowing beforehand that they exist. Thus, CATS can help shed light on low-level operational issues, and directly support knowledge transfer to practitioners via training/aiding applications. Moreover, it seems likely that data and errors that CATS detects can also help support fine-grained cognitive engineering analyses and human performance modeling research. The next sections describe issues surrounding a forthcoming attempt to demonstrate CATS using flight data from the NASA Langley B757 ARIES aircraft.

Flight Data Scope

The scope of flight data available will limit the initial test of the CATS with the B757 ARIES aircraft. An observer will record and encode clearances, as they will not be available digitally. The aircraft state parameters to be used in the test are also limited to a fraction of those used in previous simulator applications (e.g., the ACFS [23]). The data exclude interactions with the FMS—a conspicuous omission, given the increasing role of FMS-based operations in future operational concepts, and the success of CATS in tracking such activities in past demonstrations [19, 22-24]. Nonetheless, the available data are sufficient to examine pilot mode selections and target value settings on the MCP, such as the VNAV PTH-airbrake example described above.

Reasoning under Uncertainty

Great care must be taken to overcome ‘brittleness’ in a ‘pure’ rule-based framework; the above model fragment addresses, in the way it is structured, issues discussed in [14] and [15] to the extent possible. However, to address missing data and difficult-to-formulate rules for specifying context, the B757 ARIES CATS implementation will depart from the strict rule-based OFM-style modeling formulation described above. Uncertainties will instead be managed using Bayesian Network (BN)-based methods for reasoning under uncertainty. A vast body of literature espouses the value of these methods (e.g., [25-31]), and preliminary research with CATS indicates BN formulations can be processed in real-time as required for training/aiding applications.

The multiple layers of rules present in CATS lend themselves well to a hybrid approach, as rules can be thought of as BNs containing prior and conditional probabilities with only zero and one values. Thus, one possible approach is to maintain the structure of the CATS model and replace rules for generating context specifiers with BNs. Any rule whose performance is limited by uncertain data can be replaced by a BN designed to assign context specifiers the value *true* based on their posterior probabilities. This allows the CATS interpretation process to function as described above. A second approach is to employ BNs to determine when an activity is nominally predicted. A third approach is to structure the entire CATS model as a BN, with state and action nodes representing evidence variables linked to chance nodes that represent the higher-level activities operators are predicted to perform.

Of these approaches, the first is most attractive, because pilot training maintains a rule-based flavor; pilots learn heuristic rules for selecting and using autoflight modes in specific contexts, and refining these rules is an important goal of both training improvement and error-resilient system design efforts (e.g., [19, 32]). Thus, maintaining the structure of the CATS model and using it to support training/aiding while improving CATS’ capability to assess and summarize the current context holds promise for a more effective next-generation activity tracking system.

Conclusion

Airline FOQA holds great promise for enhancing safety, by using flight data to improve feedback for training. Current GDRASs support detailed analysis of data, but they do not record pilot action data or, typically, clearance and flight plan constraints. Activity tracking provides a way to help disambiguate key contextual information surrounding deviations or unusual pilot actions. If CATS has access to data that includes aircraft state, clearance constraints, and pilot actions, it can expose contextual nuances in considerable detail. Any discoveries can be incorporated into training by connecting a CATS-based training system/aid to a simulator and allowing pilots to 'fly' under conditions that, barring individual pilot differences, exactly match the actual context of the event. Such capabilities can be useful outside the airline arena (legal issues notwithstanding), as they support both fine-grained cognitive engineering analyses and human performance modeling research.

Acknowledgements

This work was funded under the System Wide Accident Prevention element of the FAA/NASA Aviation Safety Program.

References

- [1] Enders, J., 1998, FSF Study Report Urges Application of Flight Operational Quality Assurance Methods in U.S. Air Carrier Operations, *Flight Safety Digest*, Vol. 17, No. 7-9, pp. 37-46.
- [2] Prevôt, T., B. Crane, E. Palmer, and N. Smith, 2000, Efficient Arrival Management Utilizing ATC and Aircraft Automation, In K. Abbot, J. Speyer, and G. Boy, (Eds.). *HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics*, Toulouse: EURISCO, pp. 183-188.
- [3] Billings, C., 1997, *Human-Centered Aviation Automation*, Mahwah, NJ, Lawrence Erlbaum Associates.
- [4] Sarter, N., and D. Woods, 1995, How in the world did we ever get into that mode? Mode Error and Awareness in Supervisory Control, *Human Factors*, Vol. 31, No. 1, pp. 5-19.
- [5] Wiener, E. 1989, *The Human Factors of Advanced Technology ("Glass Cockpit") Transport Aircraft*, NASA Contractor Report 177528, Moffett Field, CA: NASA Ames Research Center.
- [6] Funk, K., E. Lyall, and M. Niemczyk, 1997, Flightdeck Automation Problems: Perceptions and Reality, In M. Mouloua and J. Koonce (Eds.), *Human-Automation Interaction: Research and Practice*, Mahwah, NJ, Lawrence Erlbaum Associates.
- [7] Callantine, T., C. Mitchell, and E. Palmer, 1997, Tracking Operator Activities in Complex Systems: An Experimental Evaluation Using Boeing 757 Pilots, *Proceedings of the 9th International Symposium on Aviation Psychology*, Columbus, OH, pp. 842-847.
- [8] Crane, B., T. Prevôt, and E. Palmer, 1999, Flight Crew Factors for CTAS/FMS Integration in the Terminal Airspace, *Proceedings of the 8th International Conference on Human-Computer Interaction*, Munich, Germany, pp. 1276-1280.
- [9] U.S. Federal Aviation Administration, 1995, *Advanced Qualification Program*, Advisory Circular AC 120-54.
- [10] U.S. General Accounting Office, 1998, Aviation Safety: U.S. Efforts to Implement Flight Operational Quality Assurance Programs, *Flight Safety Digest*, Vol. 17, No. 7-9, pp. 1-36.
- [11] Brandt, M., 1999, The Next Generation FOQA Programs, *International Symposium on Transportation Recorders*, Arlington, VA, May.
- [12] Anon., 1998, Aviation Performance Measuring System Software Developed, AV-DATA Aviation News, October 1998, online article.
- [13] Anon., 1999, Research and Development Plan For Aviation Safety, Security, Efficiency, and Environmental Compatibility, Cambridge, MA, National Science and Technology Council, November.
- [14] Callantine, T., C. Mitchell, and E. Palmer, 1999, *GT-CATS: Tracking Operator Activities in Complex Systems*, NASA Technical Memorandum 208788, Moffett Field, CA, NASA Ames Research Center.
- [15] Mitchell, C., 2000, Horizons in Pilot Training: Desktop Tutoring Systems, In N. Sarter and R.

Amalberti (Eds.), *Cognitive Engineering in the Aviation Domain*, Mahwah, NJ, Lawrence Erlbaum Associates.

[16] Chu, R., C. Mitchell, and P. Jones, 1995, Using the Operator Function Model and OFMspert as the Basis for an Intelligent Tutoring System: Towards the Tutor/Aid Paradigm for Operators of Supervisory Control Systems, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 7, pp. 1054-1075.

[17] U.S. Federal Aviation Administration, 1995, *Flight Operational Quality Assurance Program*, Draft Advisory Circular AC 120-XX, September.

[18] Verrière, J., 2000, FOQA Contribution to Flight Safety Management, *Flight Safety Foundation 12th Annual European Aviation Safety Seminar*, Amsterdam, March.

[19] Callantine, T., 2000, Online Analysis Of High Fidelity Simulation Data, *Proceedings of the International Conference on Cognitive Systems Engineering for Process Control 2000*, Taejon, Korea, November, pp. 175-182.

[20] Sherry, L., M. Feary, P. Polson, R. Mumaw, and E. Palmer, 2001, *A Cognitive Engineering Analysis of the Vertical Navigation (VNAV) Function*, NASA Technical Memorandum 210915, Moffett Field, CA, NASA Ames Research Center.

[21] Woods, D., L. Johannesen, R. Cook, and N. Sarter, 1994, *Behind Human Error: Cognitive Systems, Computers, and Hindsight*, CSERIAC Report 94-01, Wright-Patterson Air Force Base, OH: Crew System Ergonomics Information Analysis Center.

[22] Callantine, T., E. Palmer, and N. Smith, 1997, Activity Tracking for Precision Descent Procedure Refinement, *Proceedings of the 1997 Conference on Systems, Man, and Cybernetics*, Orlando, FL, pp. 3126-3131.

[23] Callantine, T., and B. Crane, 2000, Visualization of Pilot-Automation Interaction, In K. Abbot, J. Speyer, and G. Boy, (Eds.), *HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics*, Toulouse: EURISCO, pp 87-92.

[24] Callantine, T., 1999, An Intelligent Aid for Flight Deck Procedure Performance: The Crew

Activity Tracking System "Task Caddy," *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Tokyo, October, pp. 1095-1100.

[25] Cooper, G., R. Curry, N. Geddes, E. Horvitz, G. Wagner, and E. Palmer, 1988, *Goal-Directed Flight Path Management Systems*, Technical Report NAS2-12381, Palo Alto, CA: Search Technology.

[26] Horvitz, E., and M. Barry, 1995, Display of Information for Time-Critical Decision Making, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Montreal, August pp. 296-305.

[27] Horvitz, E., 1997, Agents with Beliefs: Reflections on Bayesian Methods for User Modeling, In A. Jamjesson, C. Paris, and C. Tasso, (Eds.), *User Modeling: Proceedings of the 6th International Conference*, Vienna: Springer, pp. 441-442.

[28] Horvitz, E., J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, 1998, Lumiere Project: Bayesian User Modeling for Inferring Goals and Needs of Software Users, *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA.

[29] Horvitz, E., A. Jacobs, and D. Hovel, 1999, Attention-Sensitive Alerting, *Proceedings of the 1999 Conference on Uncertainty and Artificial Intelligence*, Stockholm, Sweden, July, pp. 305-313.

[30] Brown, S., R. Harrington, E. Santos, and S. Banks, 1997, User Models, Intelligent Interface Agents and Expert Systems, *Proceedings of the Embedding User Models in Intelligent Applications Workshop*, June, pp. 12-17.

[31] Furuta, K. and N. Hatakeyama, 2000, Bayesian Method of Intent Inferencing Based on Human Cognitive Model, *Proceedings of the International Conference on Cognitive Systems Engineering for Process Control 2000*, Taejon, Korea, November, pp. 30-35.

[32] Javaux, D., 1998, Explaining Sarter and Woods' Classical Results: The complexity of pilot-autopilot interaction on the Boeing 737-EFIS, In N. Leveson and C. Johnson (Eds.), *Proceedings of the 2nd Workshop on Human Error, Safety, and Systems Development*, Seattle, WA.