2001-01-2640

Analysis of Flight Operational Quality Assurance Data Using Model-Based Activity Tracking

Todd J. Callantine
San Jose State University/NASA Ames Research Center

ABSTRACT

Future technologies will enable carriers to collect additional flight data for Flight Operational Quality Assurance. This paper describes how analysis of these data using model-based activity tracking can automatically assess the causes of detected deviations to support safety-enhancement efforts. The paper describes the activity tracking methodology implemented in the Crew Activity Tracking System (CATS) using an example drawn from previous research in which CATS analyzed full-mission simulation data online. The paper also discusses current research on using CATS to analyze flight data from a Boeing 757 aircraft.

INTRODUCTION

Projected air traffic growth has focused unprecedented attention on aviation safety. The attention is warranted not only because more flights create more opportunities for pilot errors—already cited as a factor in 70% of accidents (e.g., [8])—but also because the consensus solution to increased efficiency involves increased reliance on automation for accurate scheduling and precise aircraft flight trajectories (e.g., [11]). Automation, however, has a demonstrated tendency to increase cognitive workload and foster errors that can lead to unsafe operations [1, 17]. Future flight safety therefore hinges on understanding the operational contexts in which errors occur, including the role of automation. It requires methods for detecting errors and capturing error contexts, as well as methods for reducing errors and mitigating their effects.

Much of what is currently known about error-inducing contexts has been learned from analysis of accidents, simulation data, and pilot reports, such as those submitted to the U.S. National Aeronautics and Space Administration (NASA)-administered Aviation Safety Reporting System (ASRS). All of these methods have drawbacks; inexpensive, comprehensive data are required to address future safety concerns. Flight Operational Quality Assurance (FOQA) programs are an especially promising source of detailed data. FOQA programs use flight data collected during a carrier's normal line operations to reduce costs and enhance safety by identifying a variety of 'exceedances'—deviations of important flight parameters from normal operating ranges. FOQA programs typically use commercial Ground Data Replay and Analysis Systems (GDRASs) to detect specified deviations, generate plots, and create visualizations. Safety managers use this output to assess deviations that would otherwise go unnoticed, and develop prevention and remediation strategies, such as modifying the carrier's training curriculum, operational procedures, and maintenance methods [2, 8, 12, 13]. Current GDRASs offer significant benefits, but they also have a considerable weakness: without supplemental information about the error context, they provide limited information about the factors that contributed to a deviation [14].

A more objective approach is to enlarge the set of flight data to include pilot actions, Flight Management System (FMS) route database information (e.g., charted routing) and Control and Display Unit (CDU) state, as well as clearance contents. This paper adopts the perspective that, in the not too distant future, obstacles to collecting actions and detailed FMS state data can be overcome, and clearance contents will be increasingly accessible from data-linked clearance messages. With these data, airlines can perform crew-centered analyses of FOQA data in greater detail than is possible with today's GDRASs.

One method well suited for performing such analyses is model-based activity tracking. A methodology for activity tracking has been implemented and validated in the Crew Activity Tracking System (CATS) [6]. CATS uses a model of nominal crew activities to 'track' the actions crews perform. First, CATS predicts the activities that should be performed in the current flight context. Then, as pilots perform actions, CATS compares what they do to its predictions to check that crews perform the operations correctly. In some situations, various methods are acceptable; therefore CATS is also capable of determining that, although pilot actions do not match its predictions exactly, the actions are nonetheless correct.

This paper describes CATS and discusses its use for identifying and disambiguating contextual factors affecting detected deviations. The paper first describes activity tracking, then presents an example of a CATS-based analysis drawn from related research in which CATS was used for real-time analysis of crew performance in a NASA full-mission flight simulation. It then discusses safety-enhancement benefits of CATS-based analyses. Finally, the paper describes current research that seeks to use CATS to analyze data from the NASA Langley Boeing 757 (B-757) Airborne Research Integrated Experiment System (ARIES) aircraft, as part of the System-Wide Accident Prevention element of the joint U.S. Federal Aviation Administration (FAA)/NASA Aviation Safety Program.

ACTIVITY TRACKING

Activity tracking is a form of intent inference. It differs from the detection and analysis of deviations used in current FOQA programs (for a description of current FOQA, including sample data and deviation classes, see [4]). Activity tracking uses a model of nominally correct activities and a representation of the current operational context to ‘track’ operator activities and ‘understand’ that they are error-free. An activity model represents preferred methods for performing tasks, as well as other alternative, acceptable methods or techniques for achieving desired controlled system behaviors under various contexts.

The activity tracking methodology is comprised of a prediction component and an interpretation component. The former predicts the set of preferred operator activities for the current operational context; the latter interprets actual operator actions by comparing them to these expectations. Because various methods or techniques may be acceptable for accomplishing a task in some situations, the interpretation component also includes a mechanism for referencing the activity model to determine whether operator actions support an alternative, valid method for achieving the desired controlled system behavior. Besides supporting the interpretation component by generating preferred candidate activities, the prediction component is important for training/aiding applications, because it enables a training system or aid to inform operators about preferred activities in a given context.

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 [6]. Figure 1 depicts the CATS architecture and processing method, with its two threads. The prediction thread uses representations of the current state of the controlled system and constraints on its operation to derive the current operational context. The prediction thread then uses this representation to generate predictions from the activity model (①). The interpretation thread compares detected operator actions to the predicted activities (❶). 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 (❷). Figure 1 also emphasizes 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.

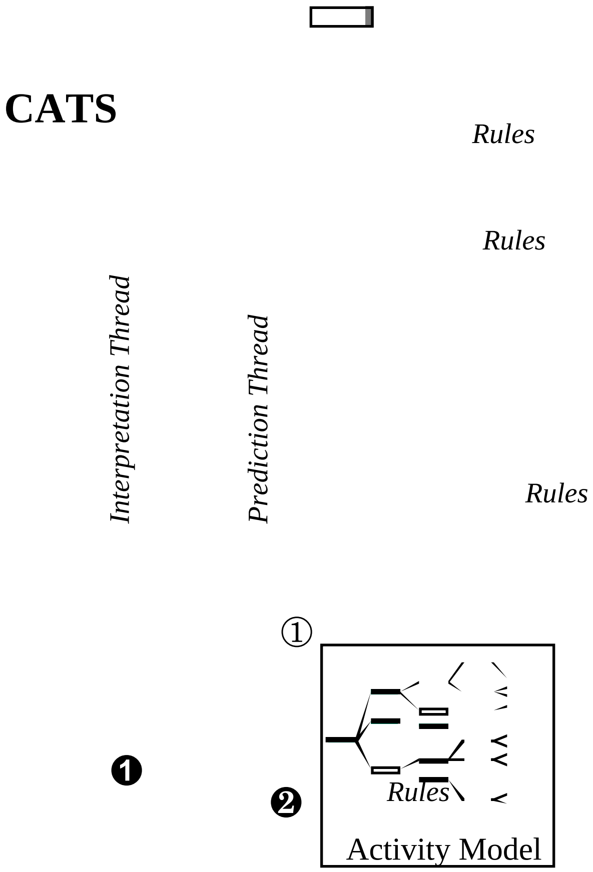
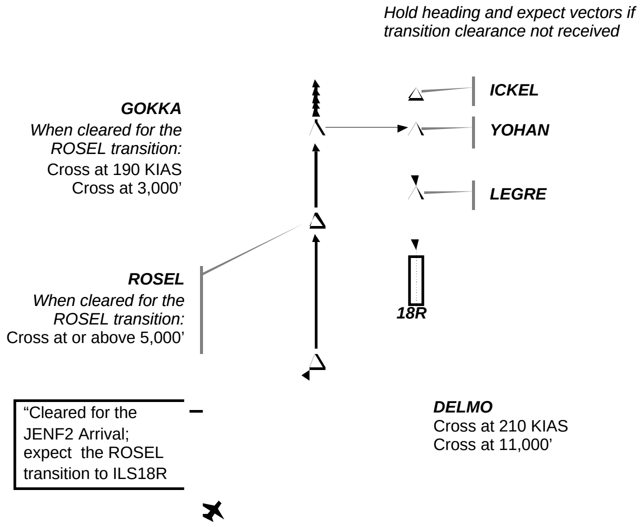


Figure 1. Generic CATS architecture and processing method.

CATS has been enhanced with visualization interfaces, like a sophisticated FOQA GDRAS [2], and used to track activities in NASA high-fidelity full-mission simulations [3, 5, 11]. Benefits realized from these research applications suggest that using CATS with actual flight data can also provide benefits. The next sections describe how an implementation of CATS for the flight deck works using an example scenario drawn from a NASA full-mission flight simulation that integrates future operational concepts [7, 10, 11]. The concepts include FMS arrivals and approach transitions that require crews to use high levels of automation in the terminal, and data linked clearances that, in addition to offering some operational benefits, enable flight data recorders to capture clearance contents for analysis. The example details the knowledge representations CATS requires, and the process CATS performs to track crew activities. It assumes familiarity with B-757-type flight deck automation.

EXAMPLE SCENARIO - Figure 2 shows a flight scenario drawn from a distributed simulation of future operational concepts with controllers in the loop using the full-motion, FMS-, data link-, and visual-equipped Advanced Concepts Flight Simulator (ACFS) at NASA Ames Research Center [12]. In the scenario, the subject crew receives the clearance "cleared for the JENF2 arrival; expect the ROSEL transition to ILS18R."



Hold heading and expect vectors if transition clearance not received
ICKEL
YOHAN
LEGRE
GOKKA
When cleared for the ROSEL transition:
Cross at 190 KIAS
Cross at 3,000'
ROSEL
When cleared for the ROSEL transition:
Cross at or above 5,000'
18R
DELMO
Cross at 210 KIAS
Cross at 11,000'
"Cleared for the JENF2 Arrival; expect the ROSEL transition to ILS18R

Figure 2. Example scenario for future FMS operations in the terminal airspace.

An important aspect of this example is that such a clearance essentially requires the use of the FMS-driven Lateral Navigation (LNAV) mode in the terminal airspace—a new, efficiency-driven operational concept. LNAV requires that the lateral route is programmed in the FMS, so the flight crew must prepare to fly the FMS transition by briefing it and programming it. However, the crew must also be prepared to revert to a lower level of automation, hold their heading, and await heading vectors if the clearance does not come by the time they reach the GOKKA waypoint (this information appears on the 'JENF2' FMS arrival chart). The preferred technique for programming the FMS in this situation is to 'load' the components of the cleared and expected flight segments, but to not 'close' the route discontinuities that result when loading the transition and approach segments from the FMS database. (The requirement for the crew to remember to close the route discontinuities to create a flyable route upon receipt of the actual transition clearance is itself error-prone and deserving of scrutiny.) This example, on the other hand, focuses on what happens when the crew 'over-commits' to the 'expect' portion of the clearance by programming the route completely ahead of time. A deviation results if the crew closes the route discontinuities and remains in LNAV *and* Air Traffic Control (ATC) is forced to withhold the transition clearance. Under these conditions, the aircraft automatically turns onto the base leg of the approach (i.e., the segment connecting GOKKA to YOHAN in Figure 2) rather than holding the current heading as required. The following sections examine how CATS tracks activities in this example to reveal errors and the need for additional training by detailing the knowledge representations and processing scheme shown in Figure 1.

State Space - CATS represents states as hierarchical whole-part relationships. Especially pertinent to the example is the state of the FMS, including the sequence of waypoints, each with its attendant latitude, longitude, and speed and/or altitude restrictions (e.g., '210 knots at 11,000 feet'). The sequence of waypoints also represents discontinuities in the route. Figure 3 shows some sample state parameters for the example.

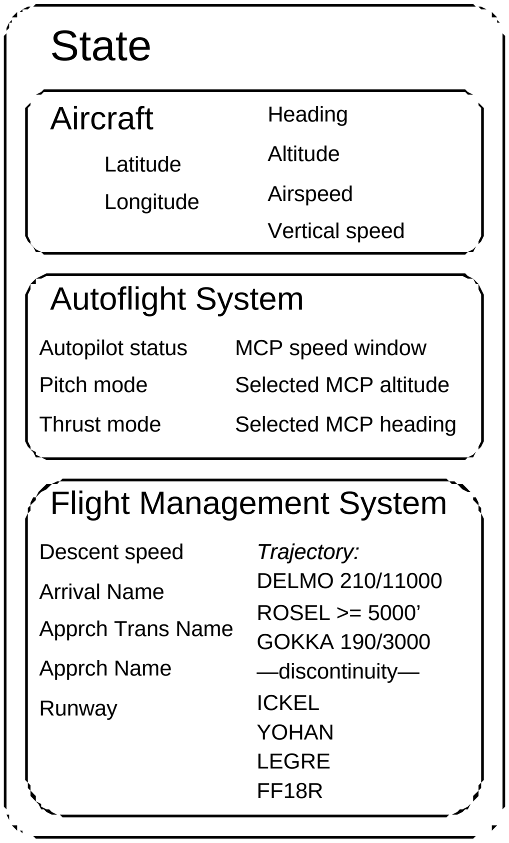


Figure 3. Partial state space for the example.

Constraints - Constraints bind the trajectory of the aircraft and include 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 'descend and maintain 5000 feet' in fact places a set of constraints on the next segment of the flight path. The constraints are those implied by the range of descent rates possible for fuel economy and 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 current 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 then become part of the complete FOQA data set. Understanding constraints on dynamic system behavior is an important part of designing and analyzing complex sociotechnical systems [10, 16]; they also play a pivotal role in activity tracking. Figure 4 illustrates constraints in effect immediately after the crew receives the clearance depicted in the example. Only the first three waypoints implied by the clearance constrain the lateral route, while only the first of the crossing restrictions constrains the vertical trajectory—the crew should expect ATC to impose the remaining constraints later. The different 'pieces' of FMS arrivals and approach transitions (i.e., waypoints, crossing restrictions, altitude limits) are yet another potential cause of confusion for both air traffic controllers and flight crews, and a subject of further research in developing future operational concepts [12]. Controllers must understand how to data link (or phrase) a clearance so as to clearly identify which elements of the charted arrival they are clearing (or overriding), while crews must understand how to manipulate the aircraft's automation to honor the correct set of constraints.

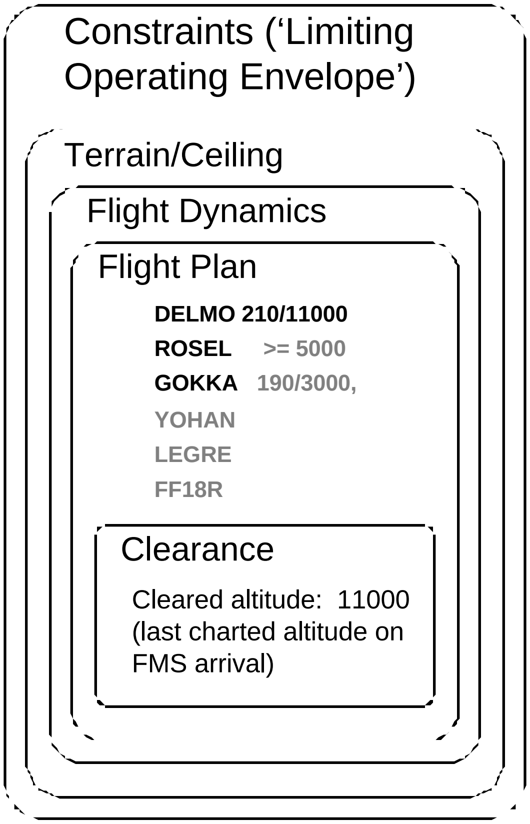


Figure 4. Example constraints; gray values are expected, but not yet confirmed.

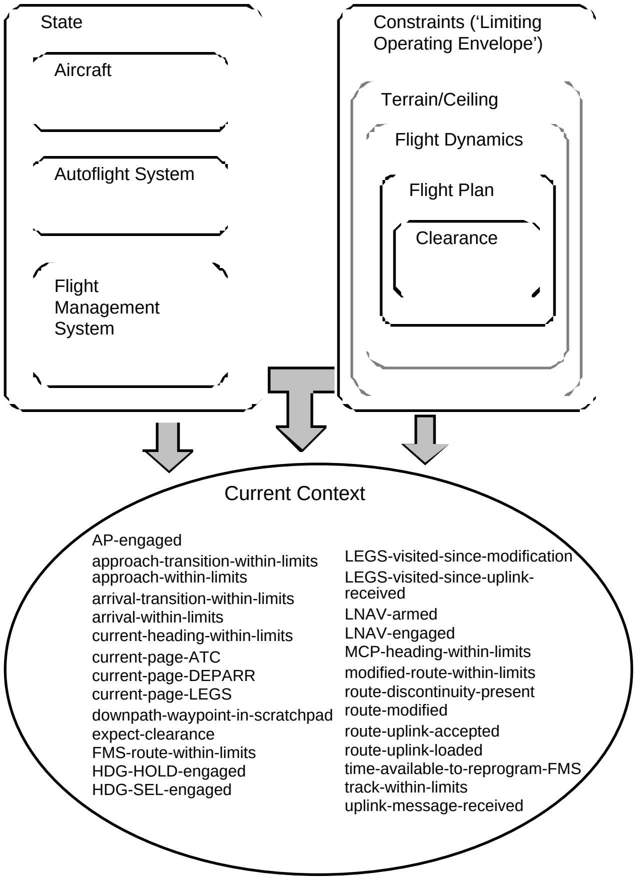


Figure 5. Context generation process, and 'context specifiers' for the example scenario.

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). 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).

Because the states and constraints form hierarchies, the context specifiers also form a hierarchy. For example, the context specifier 'FMS-route-within-limits,' which specifies that the current FMS programmed route matches that which would be required to use the FMS to meet the current constraints, subsumes the context specifier 'FMS-arrival-transition-within-limits,' because the named arrival transition is considered part of the FMS route. Each low-level context specifier has rules that express when it is `true`; CATS efficiently summarizes the current operational context by assessing the lowest level context specifiers first, and assigning the parents of those that evaluate `true` a value of `true` as well.

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 LNAV, for example. Individual pilots likely evaluate their 'mental equivalent' of such a context specifier differently, given many other 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 using the CDU. It also requires accurate predictions about the future state of the aircraft. The section on using CATS with flight data notes other methods for evaluating such context specifiers, 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. A CATS model decomposes high-level activities as necessary to adequately represent the human-machine interactions of interest, down to the level of operator actions. The hierarchical structure of the model is important not only because CATS exploits the structure when interpreting actions, but also because operators 'chunk' activities at multiple levels of abstraction, making a hierarchy useful for training. 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 (e.g., 'next waypoint') may be used in the rules for evaluating various context specifiers. This structure also distinguishes between the rules for assessing the current context, and the rules for determining what to do once the context is known. Research and training programs also sometimes distinguish between rules for 'situation assessment' and rules for 'automation management.'

Figure 6 depicts a CATS model fragment applicable to the example scenario; Table 1 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 should 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, to predict low-level activities, high-level conditions need not be repeated in them (although this practice may aid clarity). The 'memoryless' characteristic of the model—that the model can produce the nominal set of required activities for a given contextual snapshot—makes it a powerful tool for supporting 'what-if' queries. Simply adjusting the context (or the current states and constraints) produces an adjusted set of nominally predicted activities. And while the model is normative, including accurate temporal context specifiers potentially makes it more responsive to contextual subtleties (cf. [18]). A CATS 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 involved with configuring the FMS, and using the autopilot (A/P) to fly the lateral profile required by the current set of constraints. Reconfiguring the FMS involves numerous activities; the model fragment represents those related to the example, including loading a named FMS arrival, arrival transition, approach, or approach transition from the FMS database, closing route discontinuities, and executing the modified route. The model also includes the activity 'load uplink via CDU' to illustrate a representation of one data link clearance handling method. The activity 'fly A/P lateral profile' is also composed of multiple lower-level activities. These activities include setting the Mode Control Panel (MCP) target heading, using one of the available A/P roll modes to either hold the current heading, turn onto and hold an assigned heading, or tracking the FMS-programmed lateral route. The model is structured so that when the assigned heading equals the current heading, crews may select either 'heading hold' (HDG HOLD) or 'heading select' (HDG SEL) mode, with the preferred option depending upon whether they are already in HDG SEL mode or not. A detailed model includes other techniques for closing FMS route discontinuities, and making numerous other manipulations using the CDU. Portions of a complete model that represent using the autopilot to fly the vertical trajectory, configuring the flight control surfaces, manipulating the radios, etc., are omitted for parsimony here.

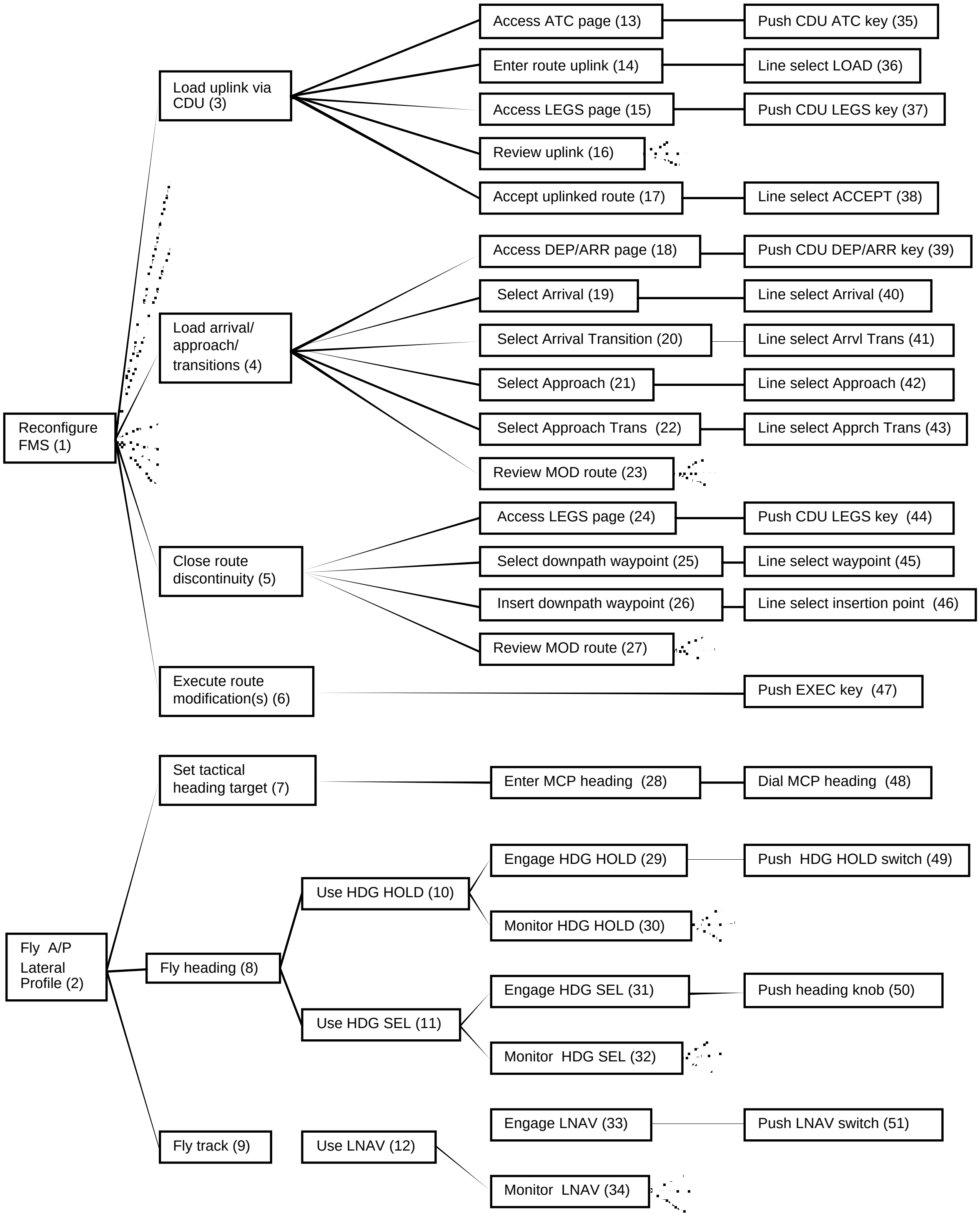


Figure 6. Example CATS model fragment.

Prediction - As shown in Figure 1, the prediction thread generates the set of activities an operator should preferably perform given the current operational context. For the example scenario and accompanying knowledge representations, CATS predicts that no further actions are required once the crew loads the expected approach transition from the FMS database and executes the modified route. In particular, the conditions for closing the route discontinuity are not met, because ATC has not yet formally cleared the 'ROSEL approach transition,' only told the crew to expect it.

Action Interpretation - 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

Table 1. Conditions for nominally predicting activities in the model fragment shown in Figure 6.

(1) (AND (NOT FMS-route-within-limits) time-available-to-reprogram-FMS)
(2) AP-engaged
(3) uplink-message-received
(4) (OR (NOT arrival-within-limits) (NOT arrival-transition-within-limits) (NOT approach-within-limits) (NOT approach-transition- within-limits))
(5) (AND (NOT expect-clearance) route-discontinuity-present)
(6) modified-route-within-limits
(7) (NOT MCP-heading-within-limits)
(8) (NOT track-within-limits)
(9) track-within-limits
(10) (AND current-heading-within-limits (NOT HDG-SEL-engaged))
(11) (OR (NOT current-heading-within-limits) HDG-SEL-engaged)
(12) track-within-limits
(13) (NOT current-page-ATC)
(14) (AND current-page-ATC (NOT route-uplink-loaded))
(15) (AND route-uplink-loaded (NOT current-page-LEGS))
(16) (AND route-uplink-loaded current-page-LEGS)
(17) (AND route-uplink-loaded LEGS-visited-since-uplink-received (NOT route-uplink-accepted))
(18) (NOT current-page-DEPARR)
(19) (AND current-page-DEPARR (NOT arrival-within-limits))
(20) (AND current-page-DEPARR (NOT arrival-transition-within-limits))
(21) (AND current-page-DEPARR (NOT approach-within-limits))
(22) (AND current-page-DEPARR (NOT approach-transition-within-limits))
(23) (AND route-modified (NOT LEGS-visited-since-modification))
(24) (NOT current-page-LEGS)
(25) (AND current-page-LEGS (NOT downpath-waypoint-in-scratchpad))
(26) (AND current-page-LEGS downpath-waypoint-in-scratchpad)
(27) (AND route-modified (NOT LEGS-visited-since-modification))
(28) (NOT MCP-heading-within-limits)
(29) (NOT HDG-HOLD-engaged)
(30) HDG-HOLD-engaged
(31) (NOT HDG-SEL-engaged)
(32) HDG-SEL-engaged
(33) (OR (NOT LNAV-armed) (NOT LNAV-engaged))
(34) LNAV-engaged
(35) (NOT current-page-ATC)
(36) (AND current-page-ATC (NOT route-uplink-loaded))
(37) (AND route-uplink-loaded (NOT current-page-LEGS))
(38) (AND route-uplink-loaded LEGS-visited-since-uplink-received (NOT route-uplink-accepted))
(39) (NOT current-page-DEPARR)
(40) (AND current-page-DEPARR (NOT arrival-within-limits))
(41) (AND current-page-DEPARR (NOT arrival-transition-within-limits))
(42) (AND current-page-DEPARR (NOT approach-within-limits))
(43) (AND current-page-DEPARR (NOT approach-transition-within-limits))
(44) (NOT current-page-LEGS)
(45) (AND current-page-LEGS (NOT downpath-waypoint-in-scratchpad))
(46) (AND current-page-LEGS downpath-waypoint-in-scratchpad)
(47) modified-route-within-limits
(48) (NOT MCP-heading-within-limits)
(49) (NOT HDG-HOLD-engaged)
(50) (NOT HDG-SEL-engaged)
(51) (OR (NOT LNAV-armed) (NOT LNAV-engaged))

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. In the example, as long as the current context reflects that the approach transition is only to be expected, not formally cleared, CATS identifies actions that support closing the FMS route discontinuity as errors. Suppose now that the aircraft is approaching GOKKA in LNAV mode, and the crew has received a clearance to descend, but not a clearance for the approach transition. At GOKKA CATS predicts that the flight crew should engage HDG HOLD mode, in order to hold the current heading as required. Should the crew fail to engage HDG HOLD, as one ACFS crew did, CATS identifies the omission after a preset time interval elapses. The aircraft continues in LNAV, and deviates from the assigned flight path by turning onto the 'base' segment of the approach.

What happens if instead ATC issues the approach transition clearance in time? Now, the 'expect' condition is gone, and the conditions for closing the route discontinuity are met. The rules for evaluating the relevant context specifiers use the variables 'downpath waypoint' and 'insertion point,' which CATS derives by comparing the desired route (represented as constraints) with the FMS-programmed route (represented in the state space). As the crew makes the required manipulations, and CATS receives updated state information, the context specifier 'modified-route-within-limits' evaluates **true**, and the conditions for predicting the 'execute route modifications' activity are met—this should be the crew's next activity.

DISCUSSION

CATS was not initially developed as an analysis tool, but rather as an intent inference system to support 'intelligent' aiding and training applications. However, in a research environment with new operational concepts under development, CATS has proven beneficial as a way to capture the designer-intended, context-specific way in which operators should use automation, and dynamically compare actual operator performance to this context-specific representation. It reduces the time necessary for high-fidelity simulation data analysis by rapidly focusing the analysis on interesting segments of the data, and captures the context of the interaction in detail [3, 5].

CATS can provide similar benefits as a FOQA analysis tool. The example above indicates the effectiveness with which analysts can use CATS to disambiguate contextual factors surrounding a deviation. Almost certainly an experienced analyst would notice the base turn deviation in the example above from a simple plot. But attributing it to a specific cause would likely require a lengthy investigation to satisfy safety managers that they have adequately addressed the possible causes of the deviation. CATS, on the other hand, immediately identifies both the error of commission (closing the route discontinuity) and the error of omission (failing to engage HDG HOLD) that led to the deviation.

Another example in a related paper [4] illustrates how CATS can also help detect erroneous activities that do not necessarily result in detectable deviations. In that example, which entailed current, rather than simulated future operations, the crew misunderstands the operation of a submode of the Vertical Navigation (VNAV) mode called VNAV Path. The crew extends their airbrakes, believing the airbrakes will increase their rate of descent, but instead this action simply causes the throttles to advance to hold speed along the VNAV path. This example indicates that CATS may also be useful for detecting 'pure' procedural deviations that need not have ever been detected before.

Model-based activity tracking has the potential to provide similar benefits for FOQA, with safety implications far beyond the oft-cited 'punitive' effect on pilots. Indeed, CATS analyses need not focus only on errors; they can also help understand what works well by examining good performance. Moreover, prerequisites to model-based activity tracking include obtaining the required data, and developing and validating the required context representations and activity models. The development and validation process places a great deal of responsibility on safety managers (especially under the current system in which each carrier independently implements its FOQA program). To develop a CATS model, implementers must specify the operational context under which each possible activity (even a single step of a procedure) is nominally preferred. To do this, implementers can define context specifiers that use their carrier's terms, and rules that reflect those taught in their training curriculum. Deficiencies in these representations do not simply limit the effectiveness or accuracy of an activity tracking system that uses them. They also highlight topics for which greater understanding is needed, and topics that need improved coverage in manuals and training curricula. In other words, managers are responsible for ensuring crews know how to fly the way the CATS says they should.

In addition, an activity tracking-based FOQA program can continue to reduce costs by directly supporting computer-based training (see [4]). As with FOQA GDRASs, the system can be integrated to work with both flight data and simulation data, so that training scenarios can be drawn from both and used interchangeably. CATS' data-replay capabilities have been used to debrief ACFS simulation subjects about their activities in a just-completed scenario; training programs could similarly benefit. Furthermore, as carriers increasingly adopt 'future' operational concepts, flight crew models can be adapted and validated to support continued training and analysis.

CATS FLIGHT DATA RESEARCH

A line of research seeks to move beyond simulators and demonstrate CATS with the NASA B-757 ARIES aircraft. An initial test seeks to demonstrate CATS' capabilities using a limited B-757 ARIES data set; subsequent tests with augmented data sets will further explore subtleties of pilot models and issues involved with using CATS for advanced FOQA analyses. In the initial test, an observer will digitally encode the time and record the contents of clearances for later insertion into the digital data stream. The aircraft state parameters to be used in the test are also limited to a fraction of those used in previous simulator applications. The data exclude interactions with the FMS, which is unfortunate considering the increasing role of FMS-based operations in future operational concepts. Nonetheless, the available data are sufficient to examine pilot mode selection and usage issues, such as the VNAV-with-airbrakes example described in [4].

The data will additionally be used to explore issues involved with data conditioning, predicting and interpreting specific classes of activities, and dealing with uncertainty due to inaccurate or missing data. The CATS modeling framework may be extended to apply Bayesian techniques in addressing these issues (see [4]). In addition to supporting activity tracking enhancements, CATS analyses and associated 'context snapshots' may also be used to support human performance modeling research.

CONCLUSION

CATS can enhance safety by using FOQA flight data to improve training feedback. CATS provides a way to disambiguate key contextual information surrounding deviations or unusual pilot actions, as well as actions that result in high performance. If CATS has access to data that includes the aircraft's 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 experience the situation.

ACKNOWLEDGMENTS

This research is funded as part of the System-Wide Accident Prevention element of the FAA/NASA Aviation Safety Program.

REFERENCES

1. Billings, C. (1997). *Human-centered aviation automation*, Mahwah, NJ, Lawrence Erlbaum Associates.
2. Brandt, M. (1999). The next generation FOQA programs, *International Symposium on Transportation Recorders*, Arlington, VA, May.
3. Callantine, T. (2000). A glass cockpit crew activity analysis tool, SAE Technical Paper 200-01-5522, Warrendale, PA: SAE International, 2000.
4. Callantine, T. (2001) The crew activity tracking system: Leveraging flight data for aiding, training, and analysis, *Proceedings of the 20th Digital Avionics Systems Conference*, Daytona Beach, FL, October, to appear.
5. Callantine, T., and Crane, B. (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, 87-92.
6. Callantine, T., Mitchell, C., and Palmer, E. (1999). *GT-CATS: Tracking Operator Activities in Complex Systems*, NASA Technical Memorandum 208788, Moffett Field, CA, NASA Ames Research Center.
7. Crane, B., Prevôt, T., and Palmer, E. (1999). Flight crew factors for CTAS/FMS integration in the terminal airspace, *Proceedings of the 8th International Conference on Human-Computer Interaction*, Munich, Germany, 1276-1280.
8. Enders, J. (1998). FSF study report urges application of Flight Operational Quality Assurance methods in U.S. air carrier operations*, Flight Safety Digest*, *17*(7-9), 37-46.
9. Klein, G., and Militello, L. (2001). Some guidelines for conducting a cognitive task analysis. In E. Salas (Ed.), *Advances in Human Performance and Cognitive Engineering Research, Vol. 1,* Amsterdam: Elsevier Science Ltd.
10. Prevôt, T., Crane, B., Palmer, E., and Smith N. (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, 183-188.
11. Prevôt, T., Palmer, E., Smith, N., and Callantine, T. (2001). Future air traffic management: A perspective on distributed automation. *Proceedings of the 8th European Conference on Cognitive Science Approaches to Process Control*, Munich, Germany, September, to appear.
12. U.S. Federal Aviation Administration (1995). *Flight Operational Quality Assurance Program*, Draft Advisory Circular AC 120-XX, September.
13. U.S. General Accounting Office (1998). Aviation safety: U.S. efforts to implement Flight Operational Quality Assurance programs, *Flight Safety Digest, 17(7-9),* 1-36.
14. Verriére, J. (2000). FOQA contribution to flight safety management, *Flight Safety Foundation 12th Annual European Aviation Safety Seminar*, Amsterdam, March.
15. Vicente, K. (1999). *Cognitive work analysis*. Mahwah, NJ: Lawrence Erlbaum Associates.
16. Woods, D., Johannesen, L., Cook, R., and Sarter, N. (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.
17. Woods, D., and Sarter, N. (2000). Learning from automation surprises and "going sour" accidents, In N. Sarter and R. Amalberti (Eds.), *Cognitive Engineering in the Aviation Domain*, Mahwah, NJ: Lawrence Erlbaum Associates, 327-353.

CONTACT

Todd J. Callantine, Ph.D., is investigating human performance modeling and model-based design methods, aids, and training systems for future aviation systems. He can be contacted at tcallantine@mail.arc.nasa.gov.