CHAP-E Demo: Monitoring Pilot Activities in the Cockpit

J. Benton, David E. Smith, John Kaneshige, Leslie Keely, and **Thomas Stucky**

NASA Ames Research Center
Moffet Field, California 94035-1000
{j.benton,david.smith,john.t.kaneshige,leslie.keely,thomas.stucky}@nasa.gov

Modern airliners have benefited from increasing automation in the cockpit. This has allowed pilots to handle more complex systems and procedures with greater safety and precision, while generally reducing workload. Unfortunately, it can also lead to misunderstanding and loss of situational awareness; pilots sometimes become fixated on the details of managing the automation and loose the overall picture of the situation, not fully comprehending what the automation will do next, or why. In the face of malfunctions or unexpected events, they must find and follow written procedures that don't take into account all the details of the particular situation, or state of the automation.

We are therefore interested in real-time cockpit monitoring of aircraft systems and flight, and providing feedback to the pilots when actions are overlooked or are inappropriate, or when the conditions of flight are no longer in accordance with the objectives or ATC clearance. Traditionally, pilots have made use of written or electronic checklists to verify that appropriate actions have been performed and that the aircraft is in the proper state for each particular phase of flight. While these have served to standardize procedures and ensure that critical items have not been overlooked, checklists are both static and passive. For example, the pre-landing checklist confirms that the flaps are at the landing setting, the landing gear is down, the airspeed is in an appropriate range for the landing weight, the approach is stabilized, etc. It does not tell the pilots when to lower flaps, when to lower landing gear, what modes and settings to select for the autopilot, or whether the selected landing speed and flap settings are even appropriate for the runway length, wind conditions, and current runway braking action. In other words, the checklist helps confirm that the state of the aircraft is appropriate, but provides no guidance about when or how to achieve that state. This information is all buried in the pilot's training and expertise, and in procedures in the Pilot's Operating Handbook (POH). However, details can get overlooked when the crew is fatigued, or overworked due to weather conditions, or system failures.

The Cockpit Hierarchical Activity Planning and Execution (CHAP-E) system is designed to assist flight crews by providing a global picture of appropriate procedures, including what actions need to be performed for the current phase of flight, and the appropriate windows for performing those actions.

In this demonstration, we will show CHAP-E monitoring a Boeing 777 on an instrument arrival and approach into San Francisco International Airport. CHAP-E shows recommended windows for each action, and alerts the pilots when actions have not been performed in a timely fashion. To do this CHAP-E requires a detailed plan for the arrival and approach that includes both hard constraints and preferences on when various actions should be performed.[1]

Figure 1 shows a small fragment of a plan for an instrument approach into San Francisco. The fragment contains three different types of statements: *events*, *actions*, and *monitors*. Each event is characterized by possible hard and soft conditions, a label, and the event itself. The event must happen within the specified hard conditions (exclamation point at the end) for the plan to remain valid. Actions must be performed by the crew and are typically keyed off of events, rather than specific times or time windows – for example, the third action specifies that the flaps must be set to 20 and the autopilot speed must be set to Vref20 between the events where the airspeed drops below Vmax20 (F20max) and crossing the defined waypoint AXMUL. There is also a soft constraint (preference) that the actions commence between crossing the intersection CEPIN and capturing the glideslope (GSCAP). If the hard constraints are met, but the soft constraints are violated, this triggers a warning or reminder to the pilots. Monitors are conditions that must hold over a given interval. As with actions, the interval is often defined by other events, such as crossing intersections, or capturing the localizer.

Figure 2 shows the CHAP-E display for some of the actions on the approach into San Francisco. At the top is a profile display of the path that the aircraft should follow on the approach, with key waypoints on the approach indicated. The aircraft is depicted as a small triangle at the upper left of this profile. Below the profile are the actions to be performed by the crew. The crew must be allowed flexibility on when to execute actions in the plan. This allows them to rely on training and preferences, while still allowing CHAP-E to provide suggested actions. The green windows indicate the recommended intervals (soft constraints),

[1] While we are working on generating these plans automatically, we have initially concentrated on the display, execution, and monitoring of these plans, and this demonstration focuses on these aspects.

```
Events {
  ARCHI: cross(ARCHI) ;
  ZILED: cross(ZILED) ;
  GIRRR: cross(GIRRR) ;
  DUMBA: cross(DUMBA) ;
  CEPIN: cross(CEPIN) ;
  AXMUL: cross(AXMUL) ;
  RW28R: cross(RW28R) ;
  before![ARCHI] {CLR: start(Clearance = ILS28R.ARCHI)} ;
  before![ARCHI] {F5max: start(IAS <= Vmax5)} ;
  before![CEPIN] {F20max: start(IAS <= Vmax20)} ;
  before![AXMUL] {F30max: start(IAS <= Vmax30)} ;
  before![DUMBA] {LocCap: start(FMA-Lateral = LOC)} ;
  before![AXMUL] {GSCap: start(FMA-Vertical = GS)} ;
  A1500: start[Alt <= 1500AGL) ;
  A1000: start[Alt <= 1000AGL) ;
  ... }
  Actions {
  between![CLR,GRRR] & between[ARCHI,ZILED] <<ArmLocalizer>> ;
  between![LocCap,AXMUL] & before[CEPIN] <<ArmGlideslope>> ;
  between![F20max,AXMUL] & between[CEPIN,GSCap]
              <<F20: SetFlaps(20),SetMCPSpeed(Vref20)>> ;
  before![A1500] & between[F20,AXMUL] <<Gear: SetGear(Down)>> ;
  between![F30max,A1000] & after[Gear] & between[AXMUL,A1000]
              <<SetFlaps(30), SetMCPSpeed(Vref30+5)>> ;
  ... }
Monitors {
  throughout[CEDES, RW28R] {IAS in [Vref,Vmax]} ;
  throughout[LocCap, RW28R] {FMA-Lateral = LOC} ;
  throughout[GSCap, RW28R] {FMA-Vertical = GS} ;
  throughout[F30, RW28R] {Flaps = 30} ;
  ... }
```

Figure 1: Fragment of a detailed CHAP-E plan for the ILS 28R approach into San Francisco

and yellow windows indicate the latest intervals that each action can be performed (hard constraints). After an execution window passes, CHAP-E can no longer guarantee that the current plan is valid for achieving the goal (a successful approach and landing in this case), and a contingency plan must be displayed. When an action is performed, CHAP-E must recognize this and remove the action from the display. This is more complex than it might first appear. For example, if the recommended action is to set the MCP-Speed to 163 knots, but the crew instead sets the speed to 165 knots, CHAP-E must determine whether this "unexpected" action still satisfies the necessary conditions for future actions that were to be achieved by the "recommended" action. If so, CHAP-E can remove the recommended action from the display. If not, CHAP-E will leave the recommended action(s), but must determine whether the unexpected action interferes with any conditions that need to be preserved in order for the plan to remain valid.

If an action is not performed (or made unnecessary by an alternative action) by the time the yellow window is reached, CHAP-E will give the crew an aural reminder. If the action is then performed before the end of the yellow window, the action is removed from the display as before. However, if the action is not completed in time, the action turns red, and the remainder of the plan is replaced by an alternative or contingency plan. In some cases this may be as simple as inserting additional actions (like the use of speed brakes). However, in general it can require more drastic revision such as aborting the approach and replacing the remainder of the approach with a plan to fly the missed approach procedure.

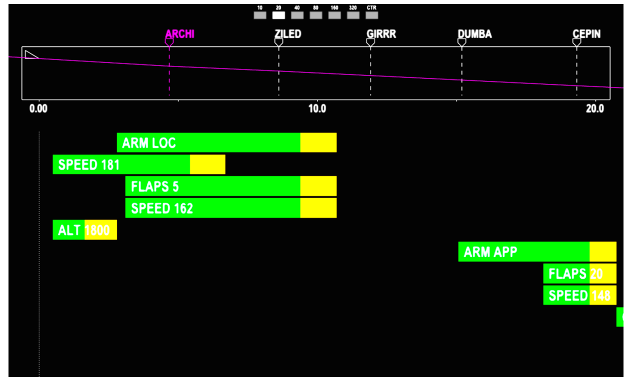

Figure 2: The CHAP-E display with approach profile above, and actions and action windows displayed below

It is also important to note that windows for actions in CHAP-E plans are not static – they can shift, shrink, or expand as a result of when previous actions are performed. For example, if the pilot reduces speed a bit later than expected, or there is less headwind than expected, subsequent actions may need to be delayed until airspeed drops into the allowed range for those actions. In order to predict this, CHAP-E must constantly reevaluate and update the earliest and latest points at which future actions can be performed. Since flight is a continuous, non-linear process, this calculation does not admit simple mathematical solution. Instead CHAP-E must constantly simulate the plan using sampling techniques to determine the windows in which actions can and should be performed. To do this, CHAP-E uses an external simulator called the Trajectory Prediction System (TPS) (Kaneshige et al. 2014). It can use these predictions to determine when a pilot may begin to execute actions in the plan and the latest time an action can be executed for the plan to remain successful. Each window is calculated independently in this process, assuming that other actions occur at the recommended time.

Acknowledgements

This work was supported by NASA Aeronautics's SASO/SECAT program.

References

Kaneshige, J.; Benavides, J. V.; Sharma, S.; Panda, R.; and Steglinski, M. 2014. Implementation of a trajectory prediction function for trajectory based operations. In *AIAA Atmospheric Flight Mechanics Conference*.