Overview of the PLEXIL Plan Execution Technology and its Applications in Autonomous Piloting Projects at NASA

Michael Dalal | Contract NNA14AA60C/NASA Ames Research Center | October 28, 2019

KBR Technology Day 2019 | Oct. 28-29 Houston, TX

Proud history, bright future.
Autonomy Operating System (AOS) for UAS

- AOS software is NASA’s Core Flight System (cFS) + AI engines
- AOS project is a feasibility study. Can AOS...

  - be an open standard for verifiable, certifiable UAS flight software?
  
  - be a reusable software platform, with associated verification technology, for UAS autonomy app development?
  
  - enable UAS to safely and reliably fly themselves in the National Air Space, behaving as a certified pilot? (“Pilot in a Box”)
“Pilot in a Box” – Technical Challenges

- Perform reasoning and decision-making needed for aviation
  - Aviate, Navigate, Communicate

- Communicate with Air Traffic Control (ATC)
  - Process clearances: understand, validate, readback, execute

- Formalize pilot procedures and execute them autonomously
  - Selected en-route, VFR/IFR approach procedures
  - Lost communication procedures
AOS Architecture

AI Apps

- Automated Reasoning (Z3, Prolog)
- Monitoring & Diagnosis (DR/R2U2)
- Autonomy Executive (DM)
- NLP (GATE)

Plan Execution (PLEXIL)

cFS

Software Bus

- Command & Mode Processor
- Actuator Manager
- State Estimator
- Safe Mode Controller
- Attitude Control System
- Thermal Control System
- Power Control System
- Battery Charge System
- Scheduler
- Stored Commands
- Memory Manager
- Memory Dwell
- Limit Checker
- Housekeeping
- Memory Scrub
- Hardware I/O
- Health & Safety
- File Manager
- CCS3S File Delivery
- Checksum
- Data Storage
- Telemetry Output
- Command Ingest

System Support and O/S Services

- Telemetry
- Gnd Cmds
- Hdw Cmds
- Sensor Data
**PLan EXecution Interchange Language (PLEXIL)**

1. **Language** for encoding plans for automation
   - Synchronous, reactive
   - Hierarchical, partially ordered plans
   - Condition and event-driven logic
   - Variety of control: concurrency, conditionals, loops
   - Formal semantics with proven properties

2. **Software** for executing plans on real or simulated systems
   - Executive (runs under Unix, Linux, embedded systems)
   - Tools: GUI, translators, checkers
Hierarchical Plans

Flight (from, to)

- FileFPlan(from,to)
- ObtainClearance
- Taxi(rnwy)
- Fly(from,to)
- Taxi(gate)
- Shutdown

- Takeoff(rnwy)
  - Climb
  - Cruise
  - Descend
  - Approach
  - Land

- CaptureLocalizer
- SetFlaps(15)
- SetMCPSpeedVref20(0)
- SetFlaps(20)
- SetGear(down)

- CaptureGlideslope
- SetMissedApprAlt
- CheckRadarAltimeterAlive
- ArmSpeedBrakes
- SetAutoBrakes(3)

- SetFlaps(30)
- SetMCPSpeedVref30(5)
- VerifyStabilizedAppr
- RunLdgChecklist
PLEXIL Formal Semantics: layered relations

- **Event:** Sigma -> Gamma
  - Sigma: world state
  - Gamma: snapshot

**Atomic**

- \((\Gamma, \pi) \vdash A.\text{start} \sim \text{true}\)
- \((\Gamma, \pi) \vdash A.\text{pre} \sim \text{true}\)
- \(A.\text{status} = \text{Waiting}\)

\[ \frac{}{(\Gamma, \pi) \vdash \text{Node A} \longrightarrow \text{Node A with status = Executing}} \]
PLEXIL Example: concurrent monitors

MonitorLoop: {
    RepeatCondition reset;
    reset = false;
    Monitor : Concurrence {
        Off_FP:
        {
            StartCondition Nominal && !Lookup(OnExpectedFlightPath);
            SkipCondition reset;
            alert (aircraft_ID, alert_ID, Caution, "Off flight path");
            Nominal = false;
            reset = true;
        }
        On_FP:
        {
            StartCondition !Nominal && Lookup(OnExpectedFlightPath);
            SkipCondition reset;
            cancel_alert (aircraft_ID, alert_ID);
            alert_ID = alert_ID + 1;
            Nominal = true;
            reset = true;
        }
    }
}
Simulated Scenario: VFR approach

nlpio done finished normally. sent TTS date to ground; see ./data/atcNL/tts/1515193242_response.txt for data sent.
EVS Port1 42/1/PLG 1: Prolog: Received PLX_MID_LAST_COMMANDED_WAYPOINT_REACHED_FROM_PLEX_MAV

--n07jb: Achieved fix slac

--n07jb: Clearance route:
1. Entry
2. Downwind
3. Base
4. Final
5. Airport

--n07jb: entering traffic pattern

--n07jb: Clearance route:
1. Entry
2. Downwind
3. Base
4. Final
5. Airport

--n07jb: Starting traffic pattern leg entry

--n07jb: Starting Direct Fix (DF) leg to midfield

--n07jb: Commanding autopilot to fly to waypoint midfield
EVS Port1 42/1/PLG 1: Prolog: fly to waypoint: sent GOTO_WAYPOINT midfield to Pixhawk
EVS Port1 42/1/MAV IB: NAV BRIDGE - received mavlink message; will send out to the pixhawk
Related Work

- Potential AOS follow-ons
  - Open sourcing of AOS
  - Collaboration with private industry
  - Incorporation on varied UAS

- Other aviation projects using PLEXIL
  - ICAROUS – open-sourced UAS architecture (NASA Langley)
  - Cockpit Hierarchical Activity Planning and Execution - CHAP-E (NASA Ames)
Thank you!

Michael Dalal
michael.dalal@nasa.gov
650.604.0053

PLEXIL home:
plexil.sourceforge.net