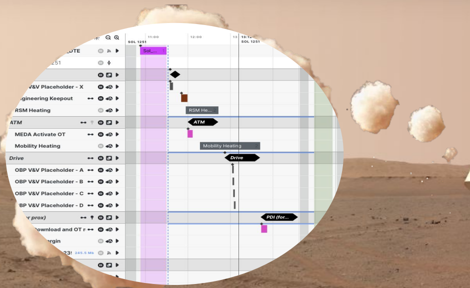
Simple Planner on Mars2020

Ground Software



Andrea Connell (she/her)

Kevin Reich, Steve Chien, Nicholas Waldram, Dan Gaines, James Biehl, James Hazelrig, Elyse Moffi, Raymond Francis

Jet Propulsion Laboratory
California Institute of Technology

February 25th, 2025

Simple Planner on Mars2020 Talk Series

Topic	Speaker	Date
Overview of Simple Planner	Moffi	5th December 2024
Onboard Planner: Flight Software	Gaines	4th February 2025
Onboard Planner: Trusted AI on Mars	Reich, Chien	18th February 2025
Simple Planner: Ground Software	Connell	25th February 2025
Simple Planner: Systems Engineering Operations with Autonomy	Hazelrig	11th March 2025
Rollout of the Simple Planner	Waldram	19th March 2025

You are here

Location: All talks are in Pickering Auditorium, Building 321, JPL Campus.
Time: All talks are 12 noon - 1 PM PST
Miss it? Recordings of all talks will be archived on JPLTube
Slides will be posted at https://ai.jpl.nasa.gov/public/projects/m2020-scheduler/

Introduction

Simple Planner is flight and ground system that enables the Mars2020 Perseverance Rover to **adjust to unexpected state**, such as Martian temperature fluctuations or battery performance and **activity execution feedback**, such as activities failing, ending earlier or later than expected. This is accomplished via scheduling autonomy onboard the rover.

Simple Planner development began in 2016, and its first use was October 5th, 2023

This talk explains the Ground Software changes that were required to allow the mission to operate effectively in the new Simple Planner paradigm.

Previous Paradigm vs Simple Planner

Master Sequence

- 11:42:03 CPU Wakeup
- 11:50:42 Manage Heating
- 11:54:30 Remote Science Submaster
- 13:05:38 Remote Science Cleanup
- 13:12:34 Drive Submaster
- 15:10:21 Drive Cleanup
- 15:19:50 Post-Drive Imaging Submaster
- 15:53:54 Post-Drive Imaging Cleanup
- 15:59:12 CPU Shutdown

Onboard Plan File

Remote Science Submaster	Time Range 11:30-13:30 Expected Duration 53 minutes Requires Mast and Camera Heating
Post-Drive Imaging Submaster	Must Start After Drive Can't be in parallel with Remote Science
Drive Submaster	Must Start After Remote Science Requires Mobility Heating Can't be in parallel with UHF Pass

- ID
- Priority
- Seq_Id
- Duration
- Avg Power
- Max Power
- Seq Engines
- Data Rate
- VDP Data Rate
- Cleanup Seq_Id
- Cleanup Duration
- Deactivate on Abort
- Activity Type
- Resource Bits
- Thermal Zones
- UHF Interaction
- Wake Interaction
- Execution Windows
  - Min Start Time
  - Max Start Time
  - Cutoff Time
  - Preferred Time
- Dependency Expressions
  - Other Activity ID
  - Execution Status

Key Takeaways

- Simple Planner was a paradigm shift requiring tight coordination between Operations, Flight Software, and Ground Software
- Many considerations to earn trust of operations team, maintain effective operability, take advantage of new capabilities, and stay within schedule and budget

Operational Considerations

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

Operational Considerations

C
consistent

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

Operational Considerations

C

U
usable

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

Operational Considerations

C

U

V

visible

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

Operational Considerations

C

U

V

P

predictable

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

Operational Considerations

C

U

V

P

A
analysis

- Maintain **consistency** with existing M20 planning process when possible
- Consider **usability** of powerful capabilities on a short tactical timeline
- Provide **visibility** into translation from planning concepts to flight software concepts
- Supply reasonable **predictability** of onboard execution when possible
- Enable **analysis** of what actually happened onboard and propagation of incoming state

C consistent

U usable

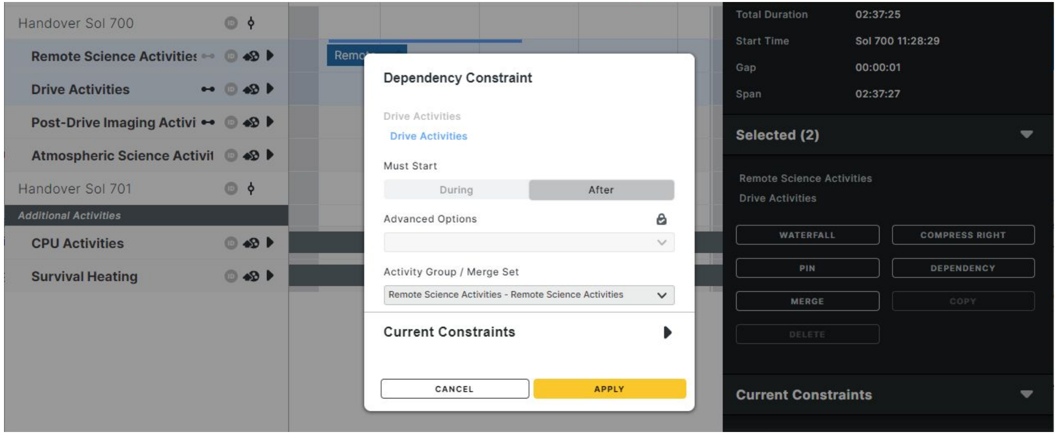
V

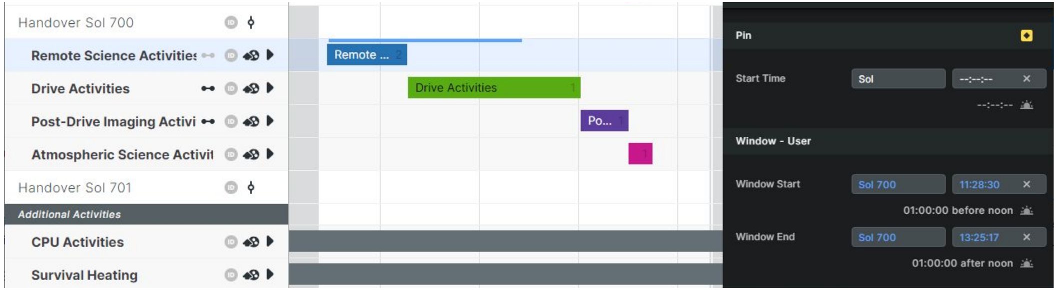
P

A

Constraint-Based Planning

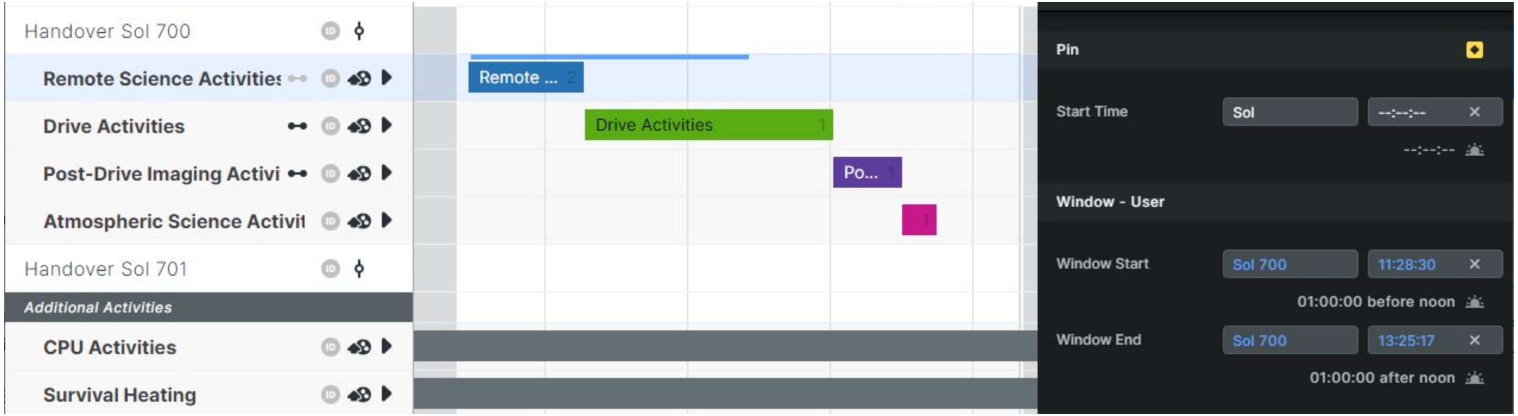
- COCPIT planning tool is used to identify scheduling constraints for activities
- Constraint concepts available at landing, and slowly built upon
- Constraint-Based Planning was used in Operations before SP rollout, and can be used with or without spacecraft autonomy





Constraint-Based Planning - Time Windows

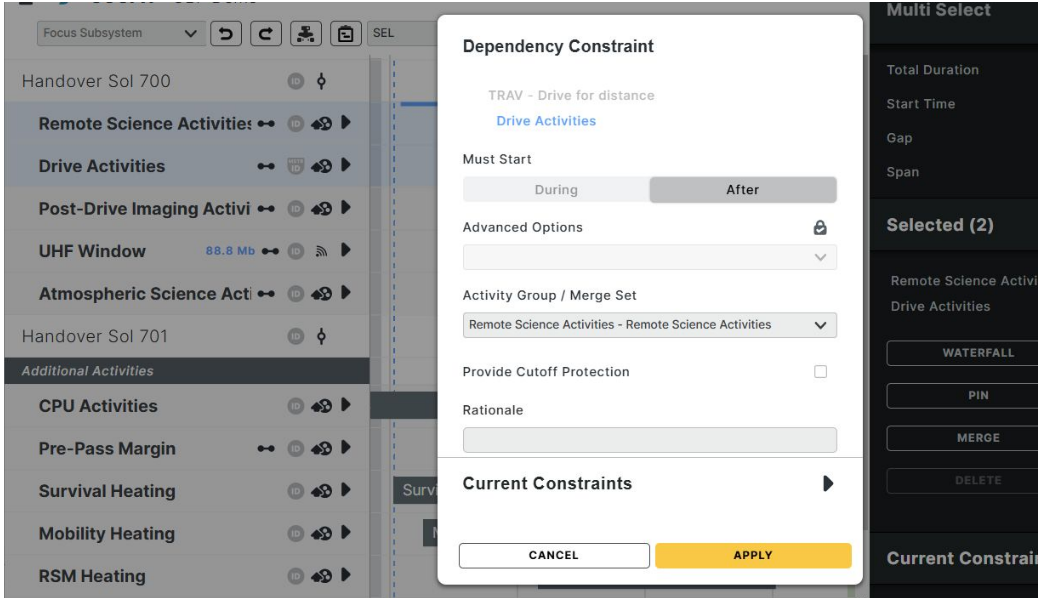
- Time windows control the range when an activity can execute
- In ground tools, these can be exact Mars time or relative to an event
- Windows can be as long or short as desired. Shortest possible window is a “pin” to a specific start time
- Allows teams to encode a valid range rather than evaluating a specific time



Constraint-Based Planning - Dependencies

C
U
usable
V
P
A

- Can set dependency constraints between activities to indicate required ordering or execution status
- FSW supports more complex constraints than we expose
- Cutoff Protection option will protect schedule time for most important activities



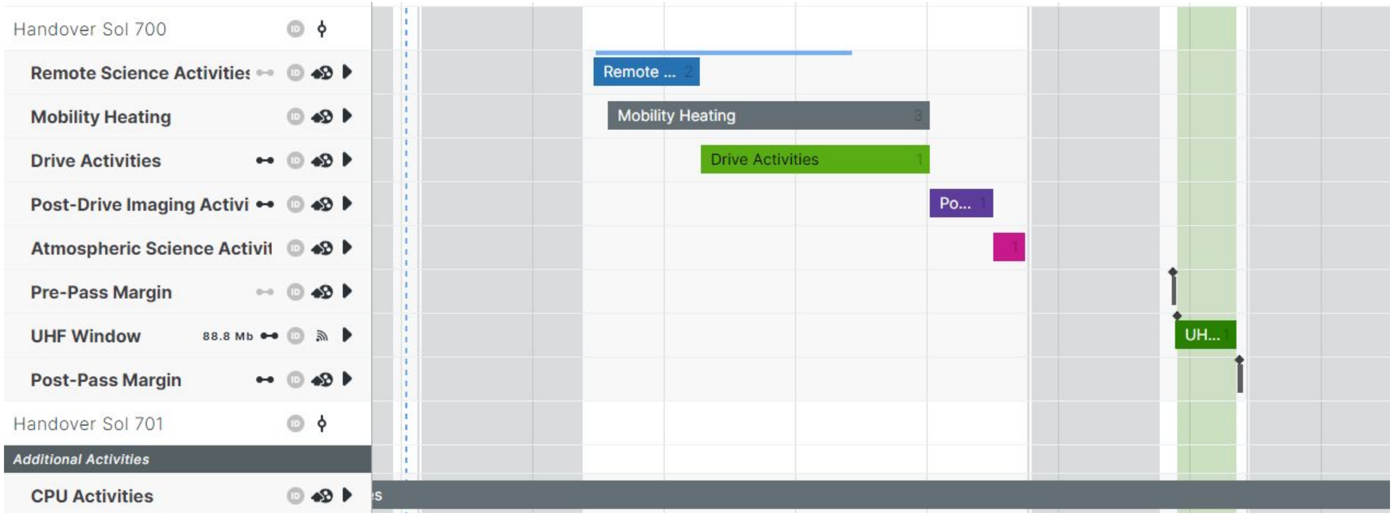
Constraint-Based Planning - Other Restrictions

C consistent
U usable
V
P
A

- Activity requires specific mechanisms to be heated before use
- Activity claims a resource, precluding parallelism with other activities that need the same resource
    - Multiple levels of parallelism restrictions encoded for onboard schedule
- Formulas to determine predicted activity duration, energy usage, and volume of data generation
- Number of sequence engines used by activity

Ground Scheduling

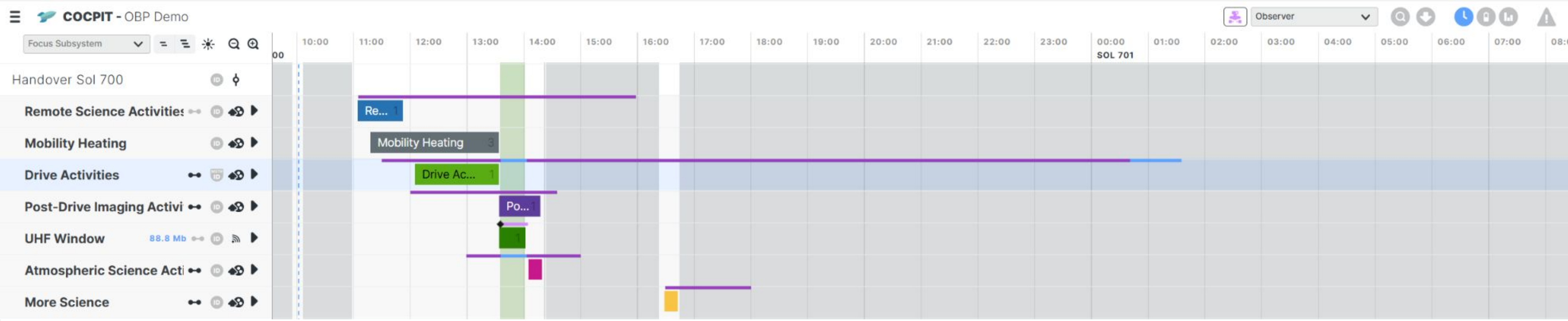
- Ground-based scheduling tool Copilot uses same underlying algorithms as FSW
- Computing power allows it to do more pre and post-processing than FSW can
- Used since landing to add CPU Wakeups/Shutdowns and Heating activities
- Now being given more flexibility via constraints



C consistent
U
V
P predictable
A

Automated Constraints to Improve Robustness

- Prevent incompatible activities from executing during comm passes
- Prevent activities from running when their mechanisms are too cold
- Provide extra heating when activities are deemed to be brittle
- Protect against nuanced FSW schedule risks



C

U
usable

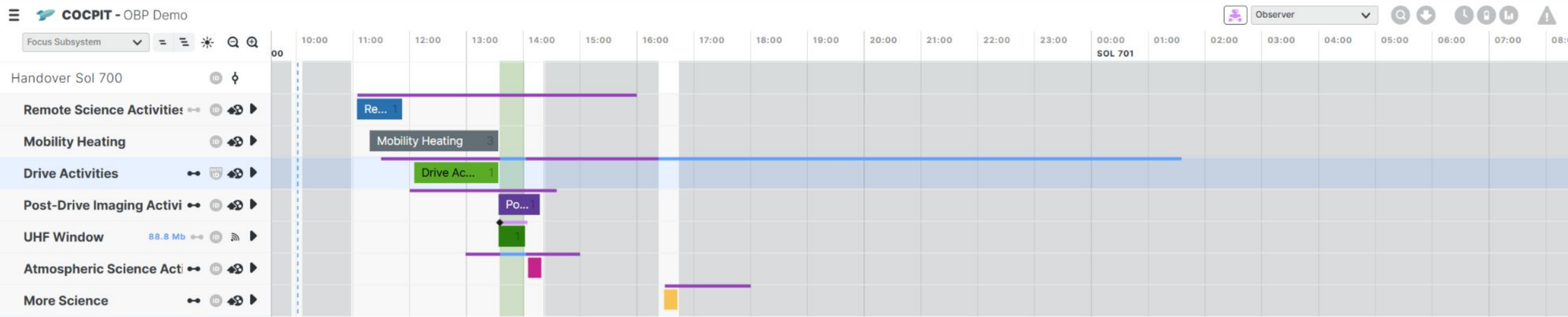
V

P

A

Automated Constraints to Ease Transition

- For initial operations, automated constraints were added to all plans
- Dependency constraints to enforce ordering and reduce parallelism
- Timing constraints to limit amount of time activities can shift
- Extra development effort was worth it for trust building and V&V phasing



C
consistent

U

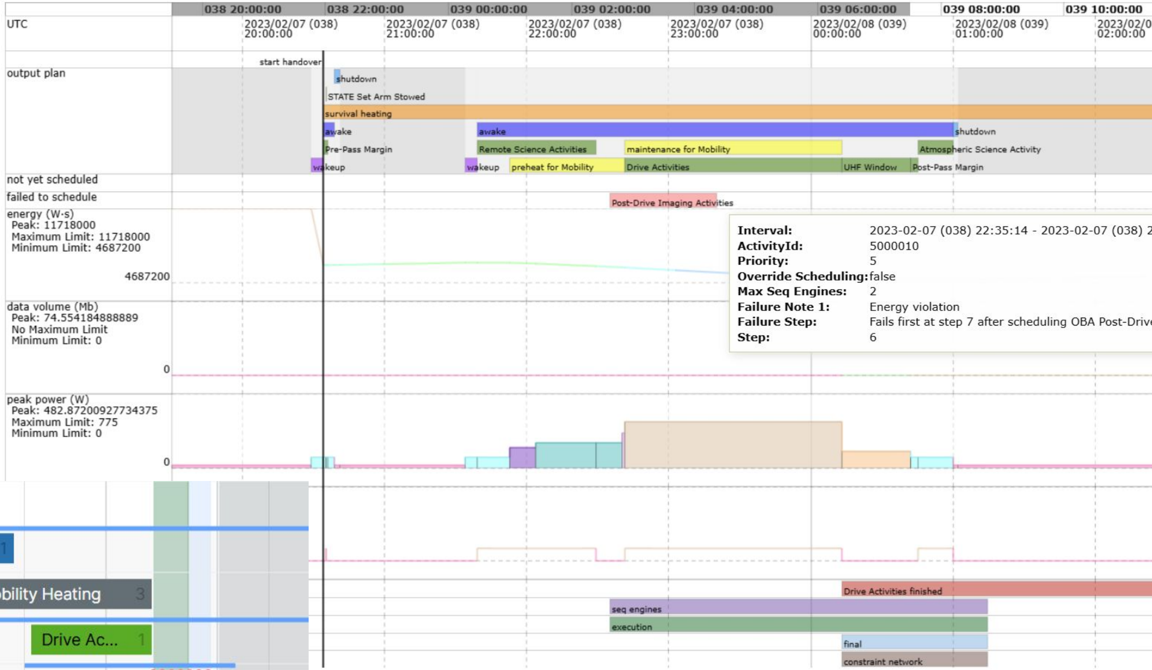
V
visible

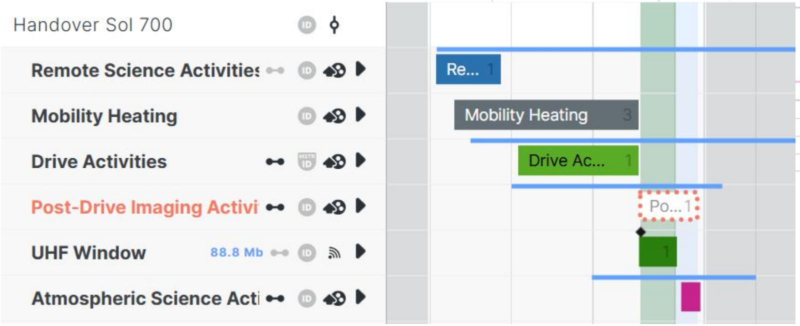
P
predictable

A

Scheduling Visualization

- Sometimes an activity cannot be scheduled
- Ideally find this in the ground tools rather than onboard
- Crosscheck tool shows ground schedule steps and explanation





Sequencing

C
consistent

U
usable

V

P

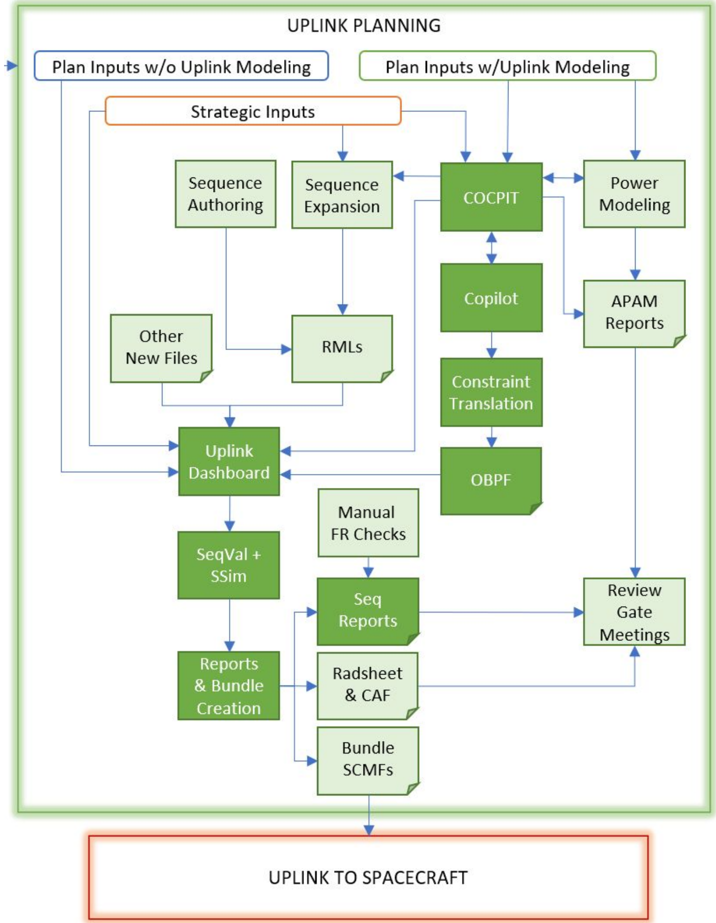
A

- Sequencing is mostly similar to previous paradigm, except Onboard Plan File replaces the Master Sequence
- Cleanup sequences only happen when an activity is cut off early
- Some sequence logic based on exact timing was adjusted
- Engineering and science teams may need to adjust their sequencing strategy to allow greater flexibility in execution time
- All previous functionality was retained for special cases

Flight Software Simulation

C consistent
U
V visible
P predictable
A

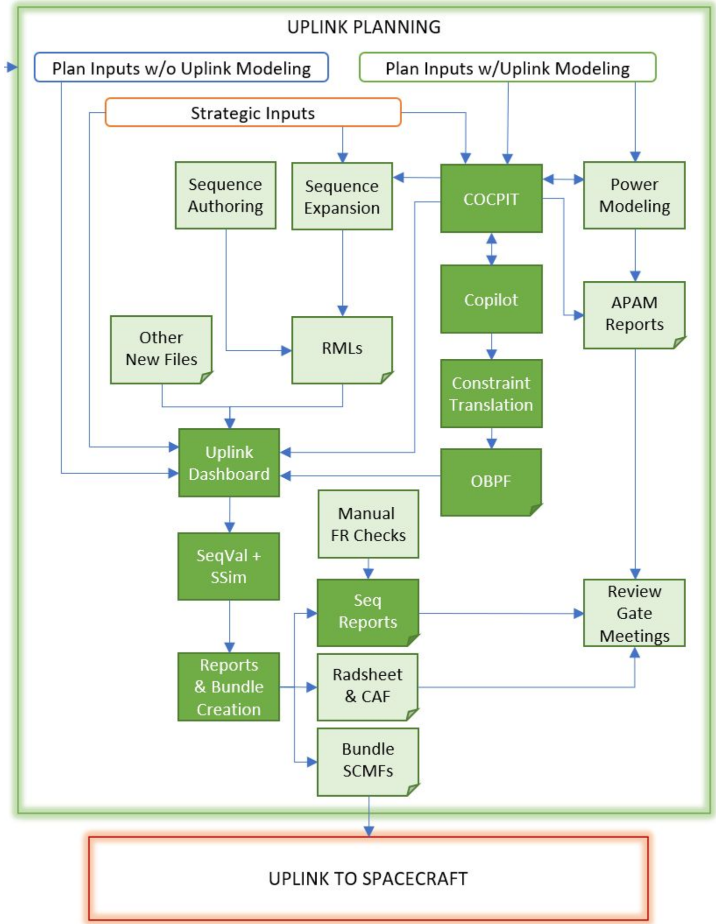
- SSim is based on real flight software modules
- Has been used since landing to simulate both paradigms
- Uses real FSW telemetry for initial state, closing the loop with propagation
- Validates Onboard Plan File
- Generates schedule for plan
- Simulates execution and rescheduling
- Provides flight-like outputs for review



Flight Software Simulation

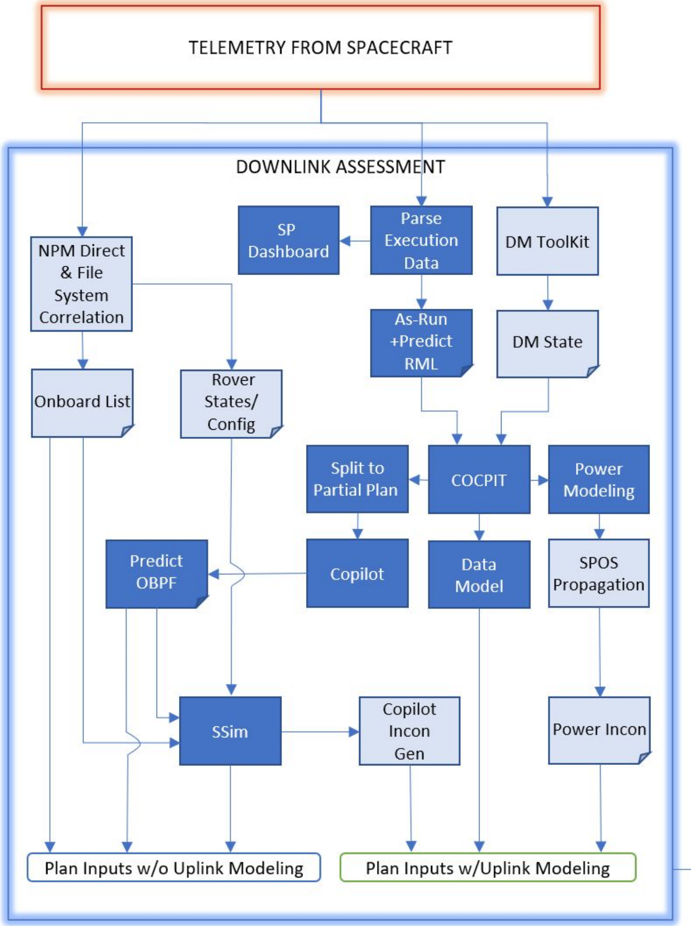
C consistent
U
V visible
P predictable
A

- Differences required in ground simulation algorithm vs flight
  - Different rescheduling triggers due to duration discrepancies
  - Energy model relies on predicted state of charge
  - Thermal interface based entirely on predictions



Telemetry Processing

- Downlink data provides visibility into onboard scheduling and execution
- Includes as-run info as well as prediction of the rest of the schedule
- New utilities needed to parse telemetry, reconstruct schedules, and correlate with uplink products for downlink assessment automation:
  - Health and safety
  - Diagnostics
  - Performance metrics
  - Activity scheduling/execution
  - Schedule visualization

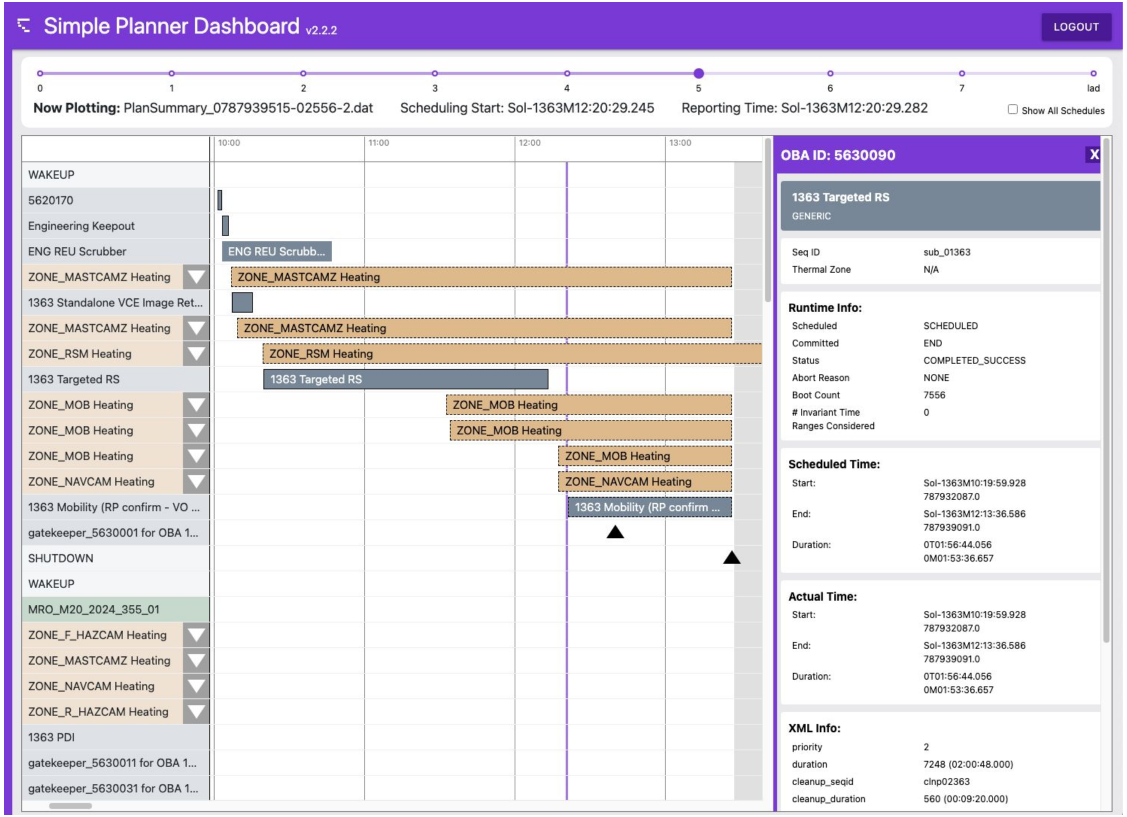


C U V P A analysis

C
U
V
P predictable
A analysis

SP Dashboard

- Visualization tool for flight and SSim telemetry
- Displays schedule states as they evolve through the day, down to the activity level
- Includes detailed diagnostics telemetry and commanded attributes



AutoRML

- Previous paradigm: Tactical plan considered sufficient for understanding plan content and activity timing
- New paradigm: Onboard schedule content and timing can vary greatly from the ground schedule
- AutoRML creates a new copy of the tactical plan with timing modified to reflect schedule based on latest downlink data
- "As-run" COCPIT plan reflects the actual schedule on the spacecraft
    - Visual aid for downlink teams to understand the actual content of the plan
    - Used to generate expected load profiles for power modeling
    - Used to bootstrap SSim with the last known schedule state

Simple Planner in Operations

- First Time Activities occurred in Spring 2023
- Primary operations mode since October 2023
- Incremental changes and improvements have continued, following normal software development cycle

Lessons Learned

- Getting consensus on the new process and earning trust in the system was often harder than the actual development work.
- More capabilities doesn’t always make for better software. Mental load of operators and ease of use also need to be considered. UX/designers and system engineers were a key part of the team.
- Operations, Flight Software, and Ground Software teams working directly together was crucial to this success. This required people with diversity of perspective and experience, who know when to challenge and when to compromise.

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov