Ames Research Center

Embedding Temporal Constraints for Coordinated Execution in Habitat Automation

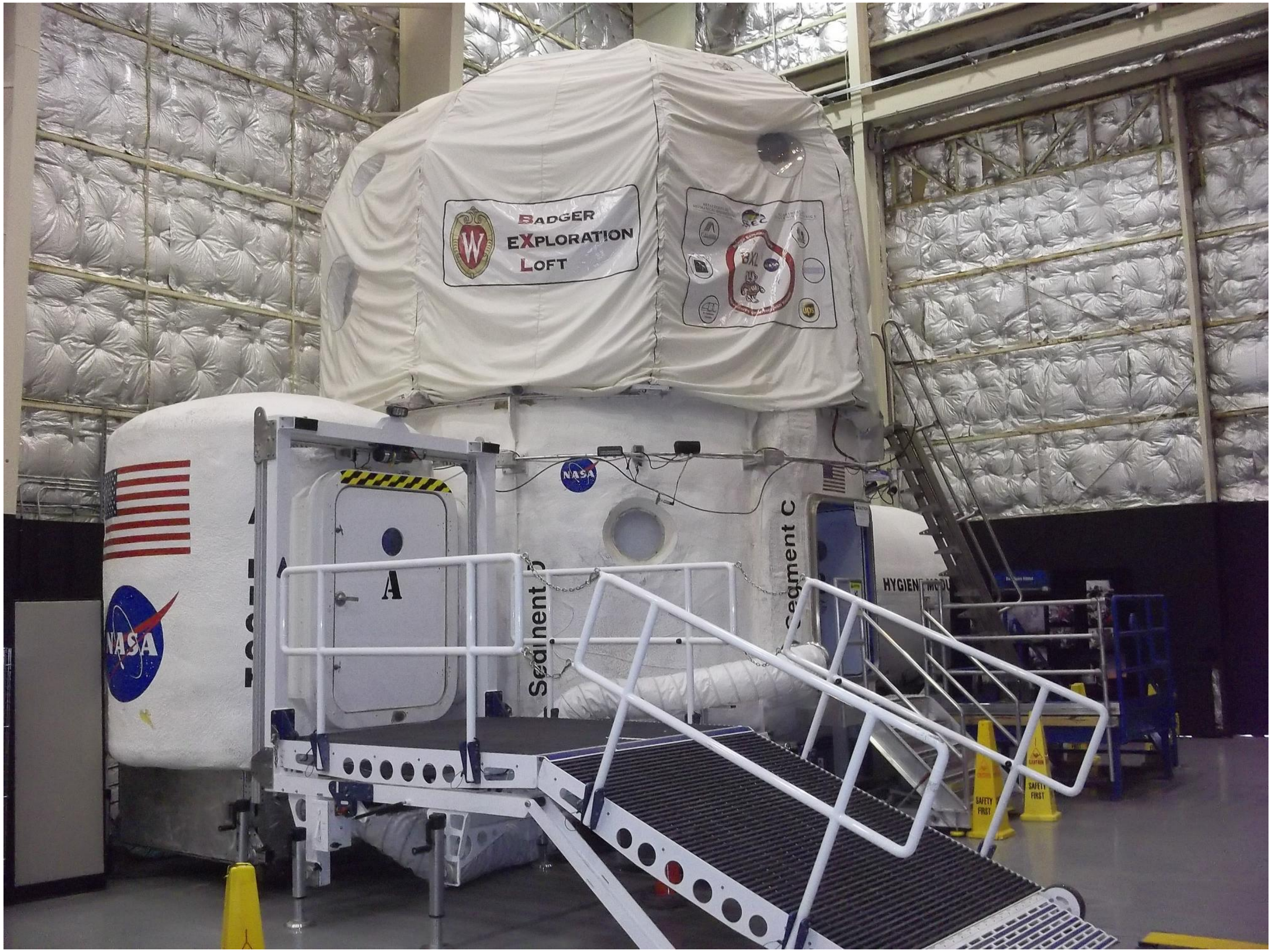
Paul Morris
Mark Schwabacher
Mike Dalal (SGT Inc.)
Chuck Fry (Dell Services)

NASA Ames Research Center

The Need

- Future deep space missions with human crews
- Long voyage, slow communication, small crew
- Partial precedents
  - Old sailing ships --- low tech
  - Nuclear submarines --- large crew
  - Remote Agent Experiment (1999) --- short duration
- Space: ~predictable, Vehicle: complex
- Need to manage vehicle behavior

Deep Space Habitat



Multiple Tool Integration

- Several Reasoning Tools

Planning/Scheduling:	SPIFe-EUROPA)
Execution	PLEXIL
Anomaly detection/diagnosis/prognosis	ACAWS/FCAS

- Need to manage interactions
- Integration experiments: explore rough edges
- Goal: re-engineer tools for smooth interactions
  - Common tool-related issues
  - Not necessarily fundamental

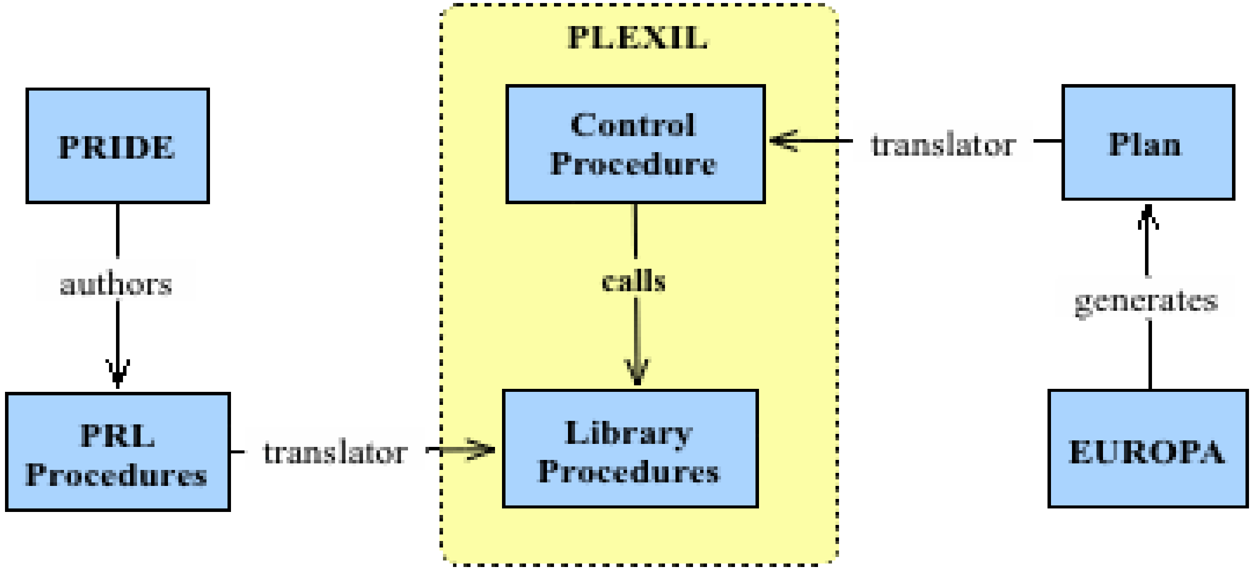
FY'12 Investigations

- Fault detection and execution
  - ACAWS (FCAS) and PLEXIL
  - Automated recovery from “routine” fault
  - Authoring of complex procedures
  - Enabled by PRL to PLEXIL translator

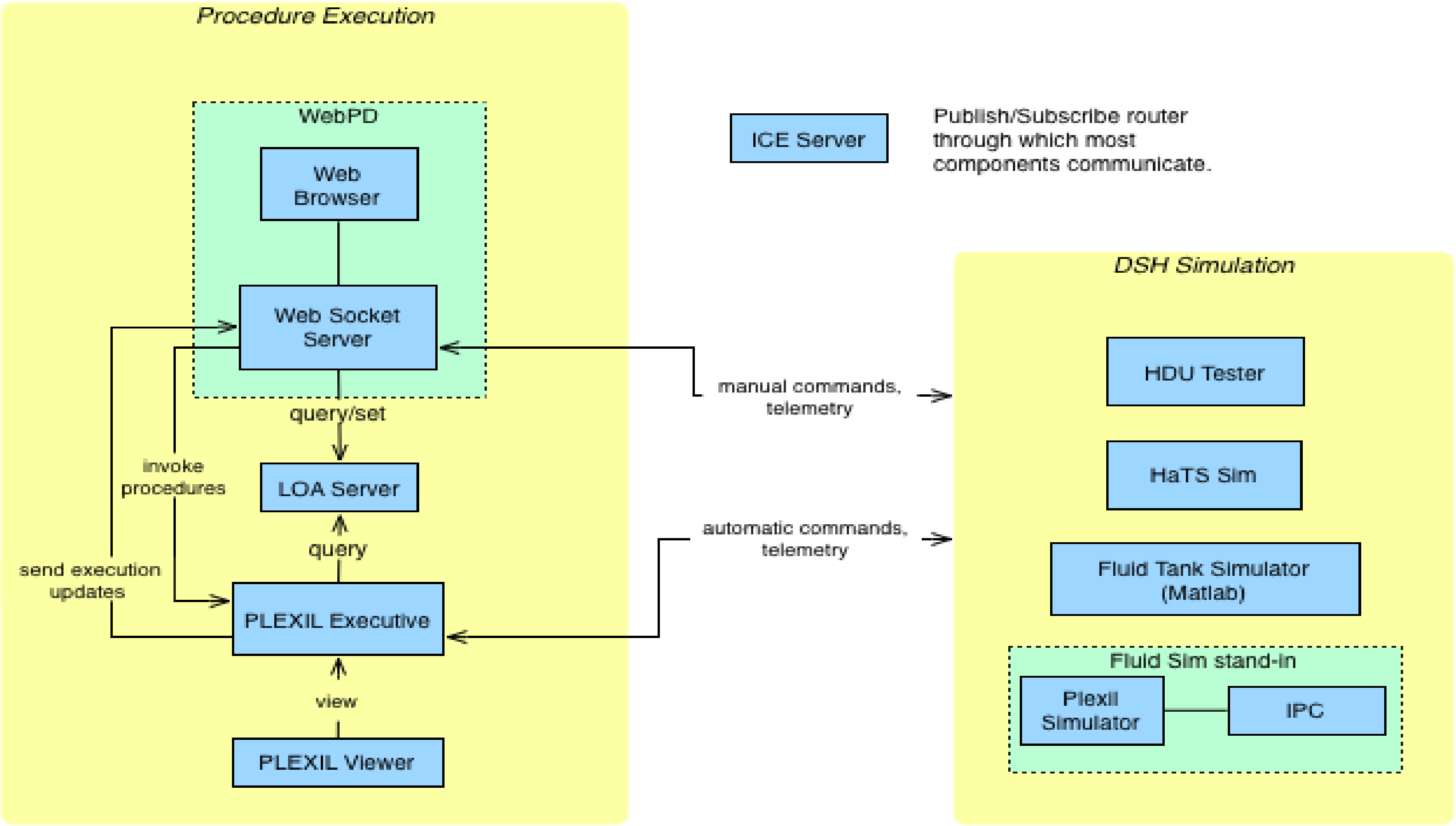
- Planning/scheduling and execution
  - Embed temporal constraints into PLEXIL procedure
    - Waits, deadlines, coordination
  - Flexible execution (adaptive scheduling, skipping)
  - Enabled by EUROPA to PLEXIL translator

Conceptual Combination

- PRL to author low-level detailed procedures
- EUROPA-generated procedure for high-level control



Runtime Software Architecture



Scenario

- Low-priority interrupted by high-priority
  - Routine camera survey repeated daily
  - Urgent fluid transfer (preempts survey)
  - If short duration, resume survey, otherwise skip

- Flexible sequence from Planner

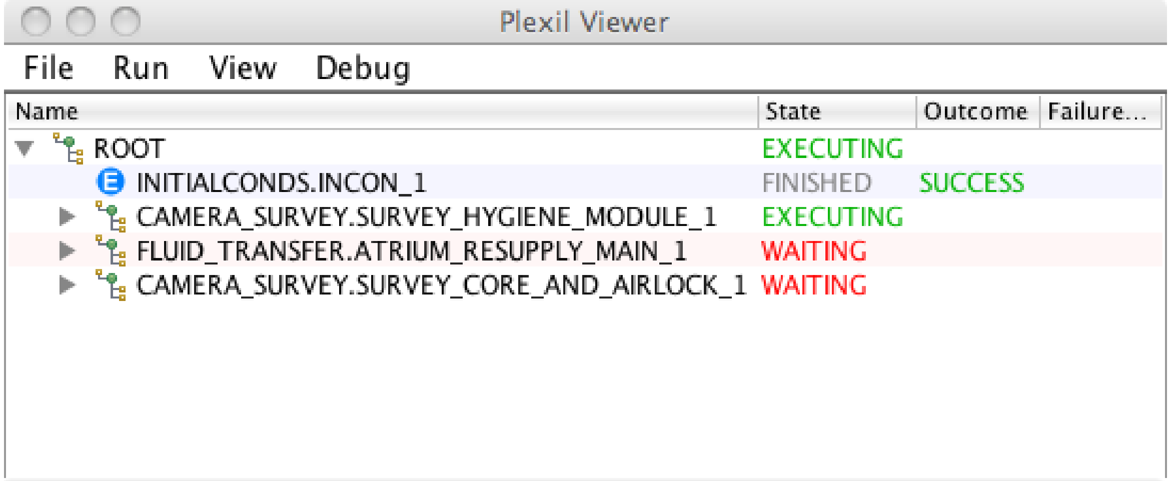
  (1) camera survey I; (Hygiene_Module)

  (2) **fluid xfer**;

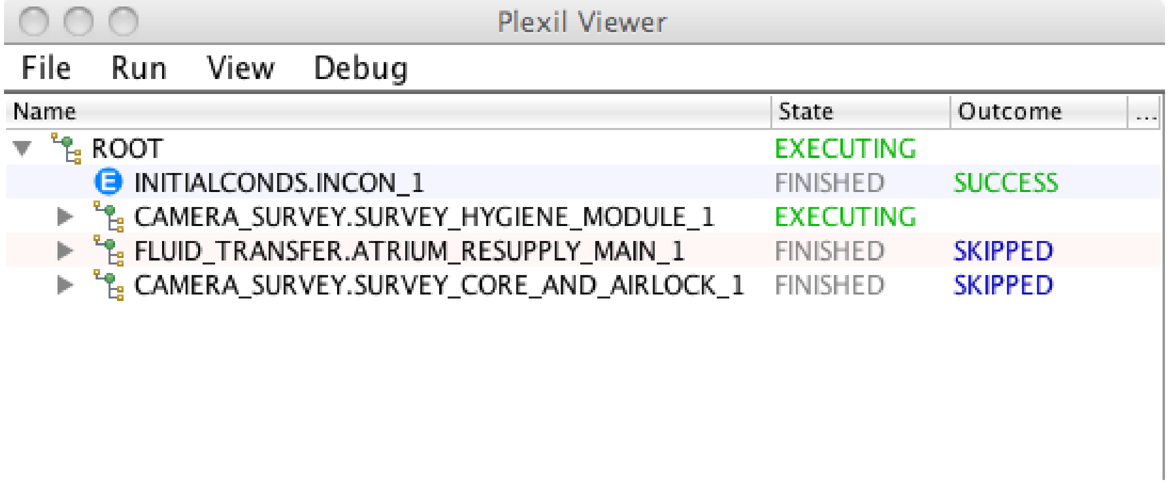
  (3) camera survey II (Core_and_Airlock)

- Nominal times, precedences, deadlines

PLEXIL Viewer

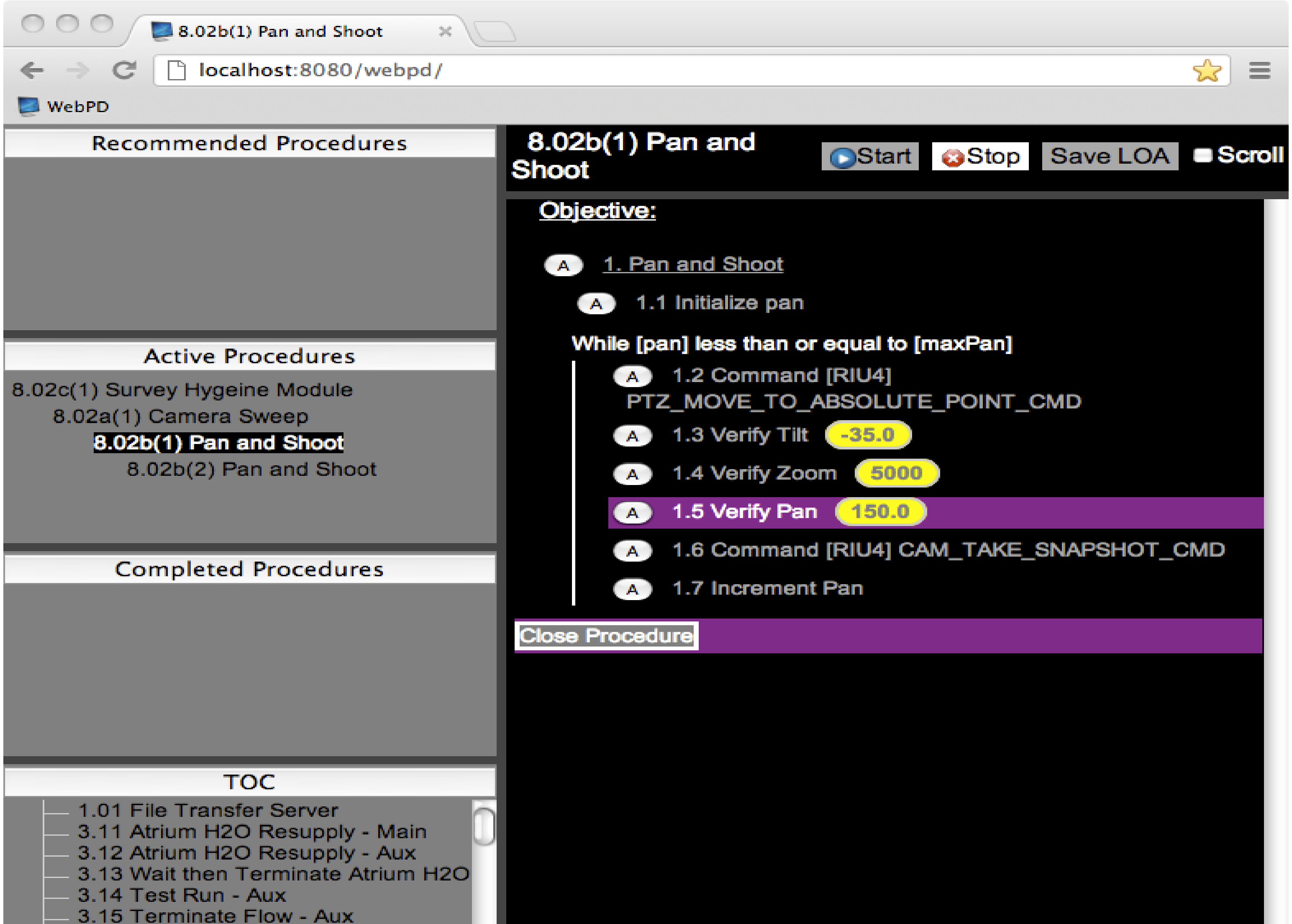


Name	State	Outcome	Failure...
ROOT	EXECUTING		
INITIALCONDS.INCON_1	FINISHED	SUCCESS	
CAMERA_SURVEY.SURVEY_HYGIENE_MODULE_1	EXECUTING		
FLUID_TRANSFER.ATRIUM_RESUPPLY_MAIN_1	WAITING		
CAMERA_SURVEY.SURVEY_CORE_AND_AIRLOCK_1	WAITING		



Name	State	Outcome	...
ROOT	EXECUTING		
INITIALCONDS.INCON_1	FINISHED	SUCCESS	
CAMERA_SURVEY.SURVEY_HYGIENE_MODULE_1	EXECUTING		
FLUID_TRANSFER.ATRIUM_RESUPPLY_MAIN_1	FINISHED	SKIPPED	
CAMERA_SURVEY.SURVEY_CORE_AND_AIRLOCK_1	FINISHED	SKIPPED	

WebPD Viewer



Recommended Procedures

Active Procedures

8.02c(1) Survey Hygeine Module
8.02a(1) Camera Sweep
8.02b(1) Pan and Shoot
8.02b(2) Pan and Shoot

Completed Procedures

TOC

- 1.01 File Transfer Server
- 3.11 Atrium H2O Resupply - Main
- 3.12 Atrium H2O Resupply - Aux
- 3.13 Wait then Terminate Atrium H2O
- 3.14 Test Run - Aux
- 3.15 Terminate Flow - Aux

8.02b(1) Pan and Shoot

Objective:

A 1. Pan and Shoot

A 1.1 Initialize pan

While [pan] less than or equal to [maxPan]

A 1.2 Command [RIU4] PTZ_MOVE_TO_ABSOLUTE_POINT_CMD

A 1.3 Verify Tilt -35.0

A 1.4 Verify Zoom 5000

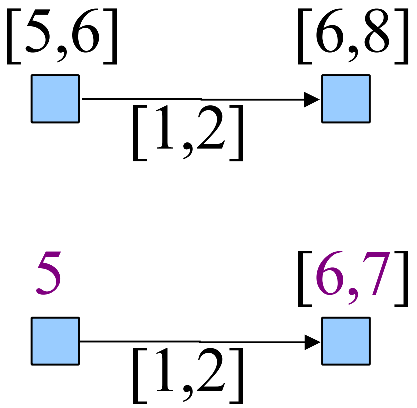
A 1.5 Verify Pan 150.0

A 1.6 Command [RIU4] CAM_TAKE_SNAPSHOT_CMD

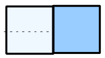
A 1.7 Increment Pan

Planner/Execution Integration Issues

- Planner has flexible times (STN)
  - Upper/lower bounds for each activity
  - Constraints between activities
  - Earlier executions affect later bounds
  - How to update? (No STN in PLEXIL)



- Plan assumes nominal durations
  - But durations may vary during execution
  - How to maintain integrity of constraints?

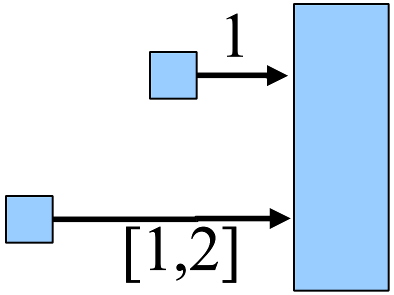


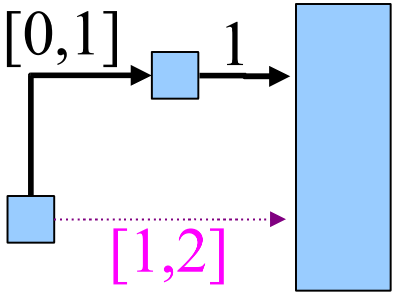
Approach

- Convert plan to *Minimum Dispatchable* form
  - Need only *local* propagation

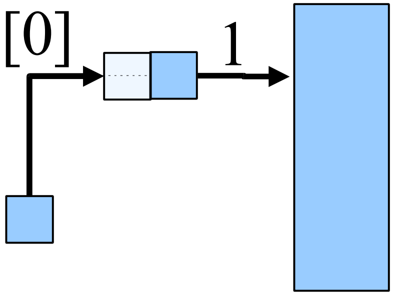
    = additional bounds **relative** to neighbors

    end(A)+[0,1], end(B)+[5,7], ...
  - Few neighbors (minimum form)





- Apply *Dynamic Controllability* algorithm
  - Models temporal uncertainty
  - Transfers flexibility to manage uncertainty



Approach, *cont.*

- Lower, upper bounds → PLEXIL *start* conditions
  - Wait for *nominal* time **...**

    ... unless about to hit a *deadline* (upper bound)
  - Must wait for *lower bounds*, regardless of deadlines
  - time ≥ max[lb, lb1, lb2, ... , min[*nom*, ub, ub1, ub2, ...]]

Approach, *cont.*

- Lower, upper bounds → PLEXIL *start* conditions
  - Wait for *nominal* time **...**

    ... unless about to hit a *deadline* (upper bound)
  - Must wait for *lower bounds*, regardless of deadlines
  - time ≥ max[lb, lb1, lb2, ... , min[*nom*, ub, ub1, ub2, ...]]
- Upper bounds → PLEXIL *skip* conditions
  - time ≥ min[ub, ub1, ub2, ...]] + *latency*

Approach, *cont.*

- Lower, upper bounds → PLEXIL *start* conditions
  - Wait for *nominal* time **...**

    ... unless about to hit a *deadline* (upper bound)
  - Must wait for *lower bounds*, regardless of deadlines
  - time ≥ max[lb, lb1, lb2, ... , min[*nom*, ub, ub1, ub2, ...]]
- Upper bounds → PLEXIL *skip* conditions
  - time ≥ min[ub, ub1, ub2, ...]] + *latency*
- Conditions not satisfied until *known* to be satisfied

  e.g. lb1 = end(A)+20 unknown if A not finished

Lessons Learned

- Skip *iteration* issues (smarter skipping)
  - Skip follower: precedence from **precond** or **mutex**?
  - Predicted skips
    - Obsoleted precond activities
    - Obsoleted inferred deadlines
- Options
  - Embed causal/support information
  - Group activities
  - Round-trips to planner

Future Work

- Integrate Diagnosis/Replanning
  - ACAWS diagnostic agent identifies failed components
  - Planner determines activity impacts, repairs plan
- Three-way integration
  - Execution/diagnosis/replanning
- Plan repair spectrum
  - Reordering, postponing, abandoning
  - Resource switching, alternate methods
  - Novel combinations of primitives (creative, trust issue)
  - Remodelling (ultra-creative, c.f. Watson)

Backup Slides

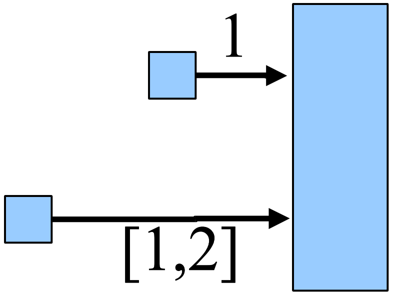
Products/Milestones

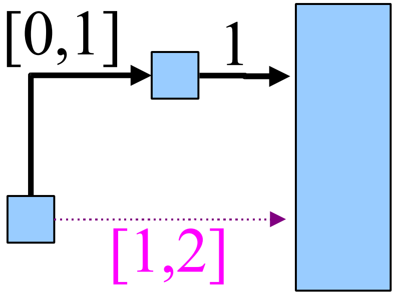
- PRL to PLEXIL translator
  - Bug fixes
- EUROPA to PLEXIL translator (new)
- WebPD upgrades
  - Iteration
  - Direct user typed input
  - Bug fixes
- PLEXIL upgrades
  - ISO 8601 dates, times, durations

Approach

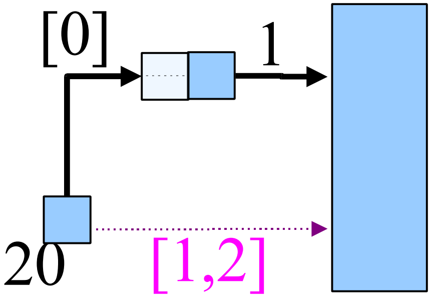
- Convert plan to *Minimum Dispatchable* form
  - Need only *local* propagation

    = additional bounds relative to neighbors
  - e.g., end(A)+[0,1], end(B)+[5,7], ...
  - Parsimonious network



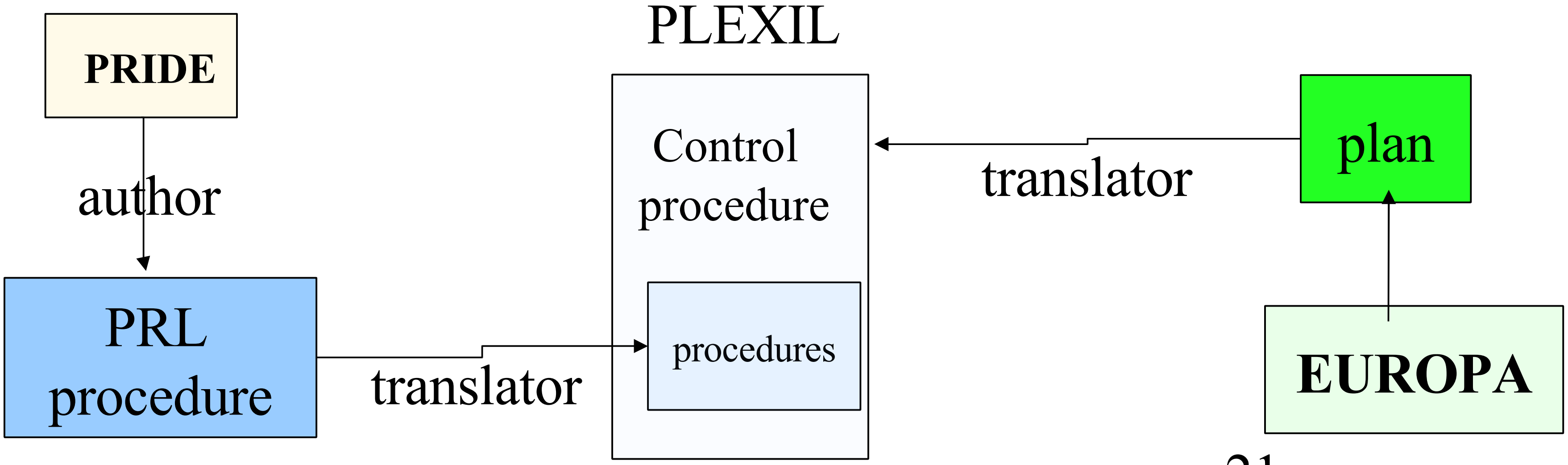


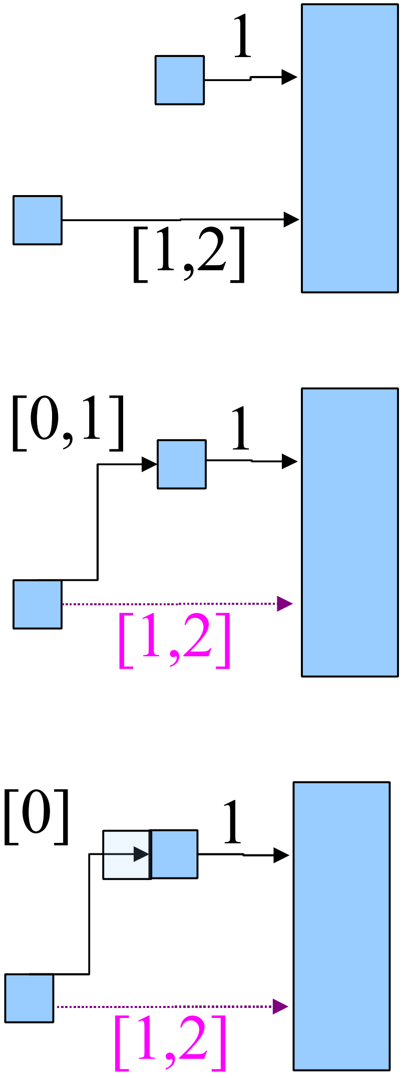
- Apply *Dynamic Controllability* algorithm
  - Models temporal uncertainty
  - Transfers flexibility to manage uncertainty



Conceptual Combination

- PRL to author low-level detailed procedures
- Europa-generated procedure for high-level control




1
[1,2]
[0,1]
1
[1,2]
[0]
1
[1,2]