Practical Planning
Part I

Ari Kristinn Jónsson
Reykjavik University

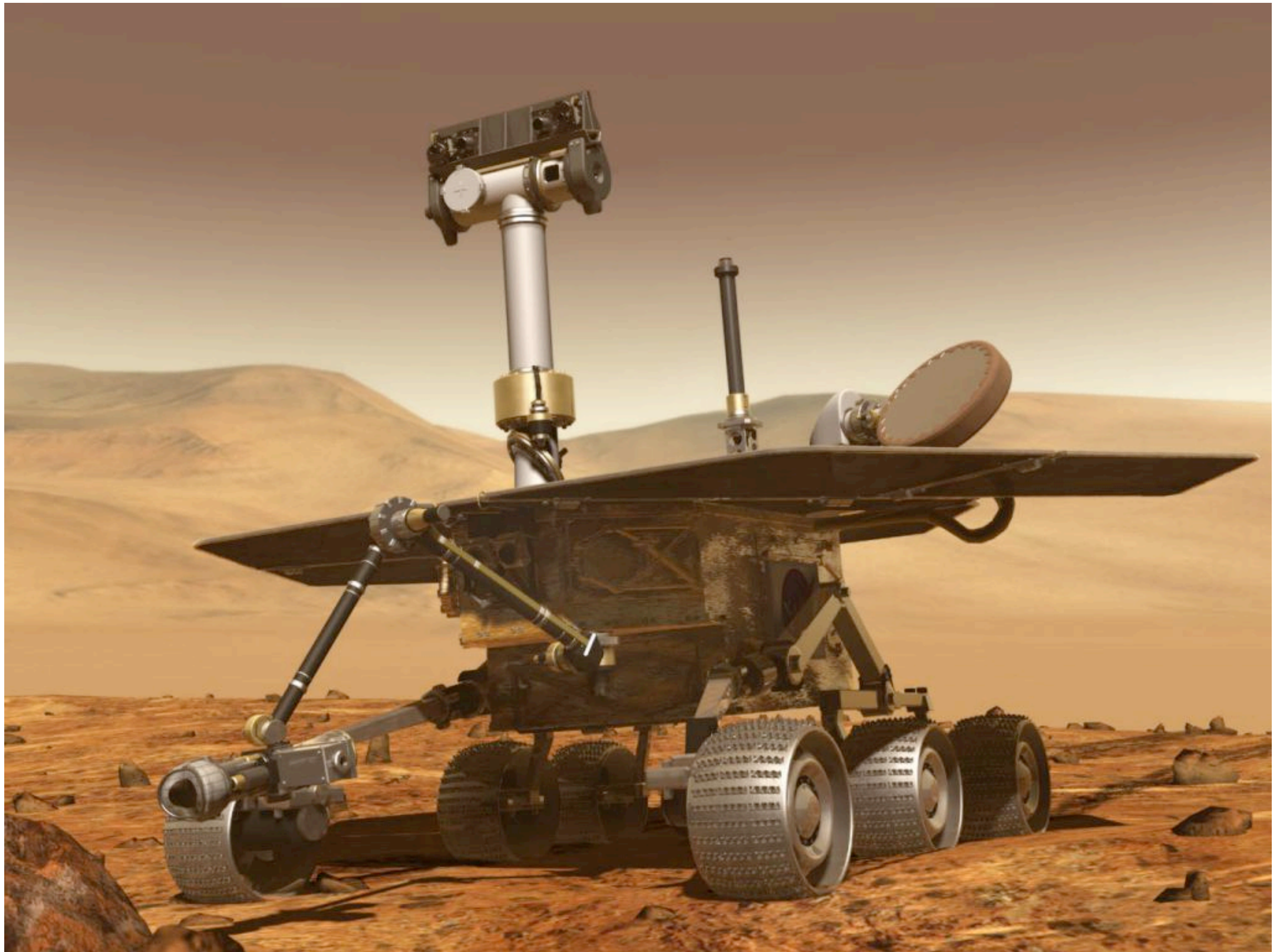
Content

- Requirements of "practical planning"
  - What is different from classical planning?
- Example application: Mars Exploration
  - What might practical planning look like?
- Constraint-based planning
  - What kind of planning might be used?
- Developments and future applications
  - What will we see next in practical planning?

Let’s start with a motivating example

MARS EXPLORATION ROVERS

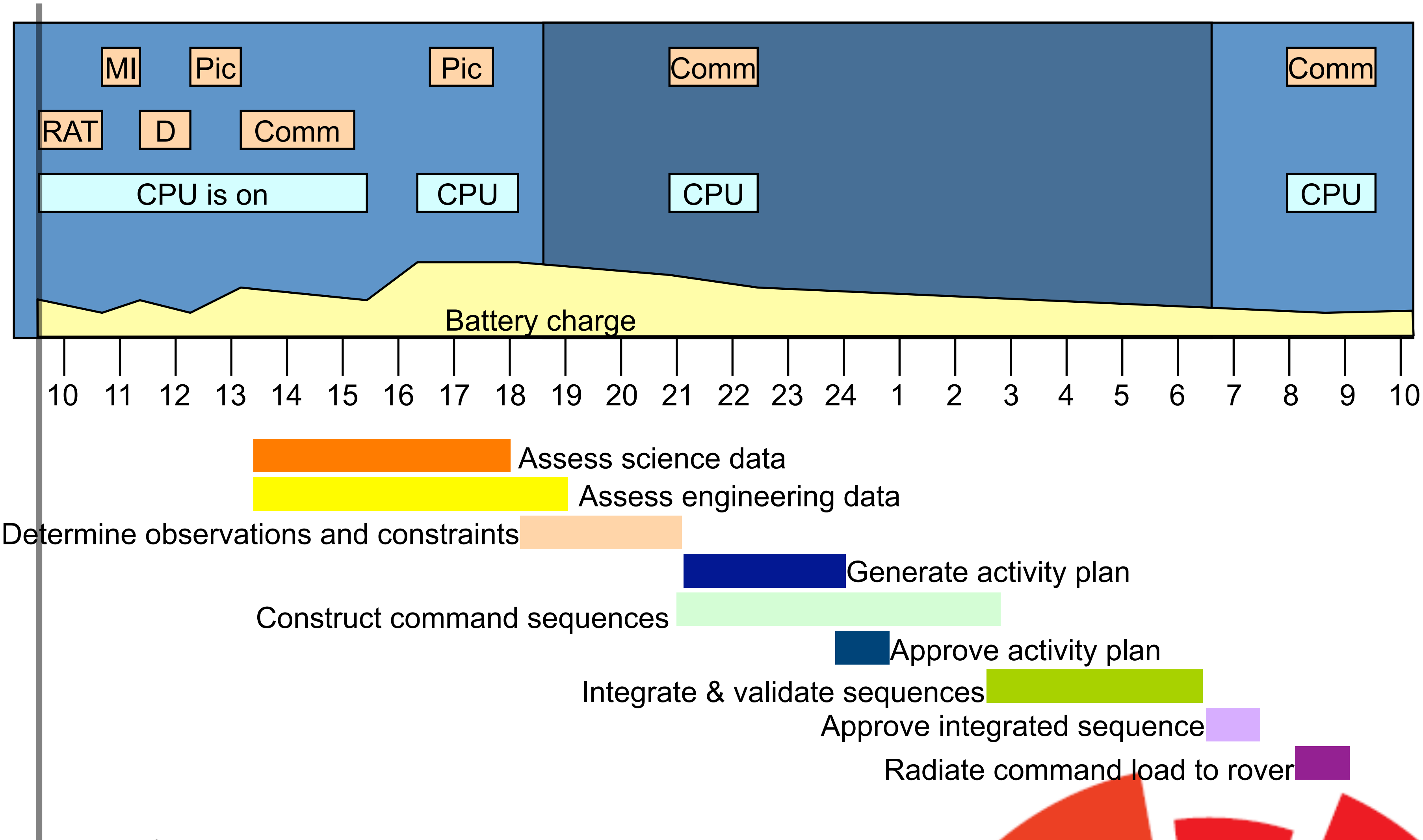
Mars Exploration Rover (MER)



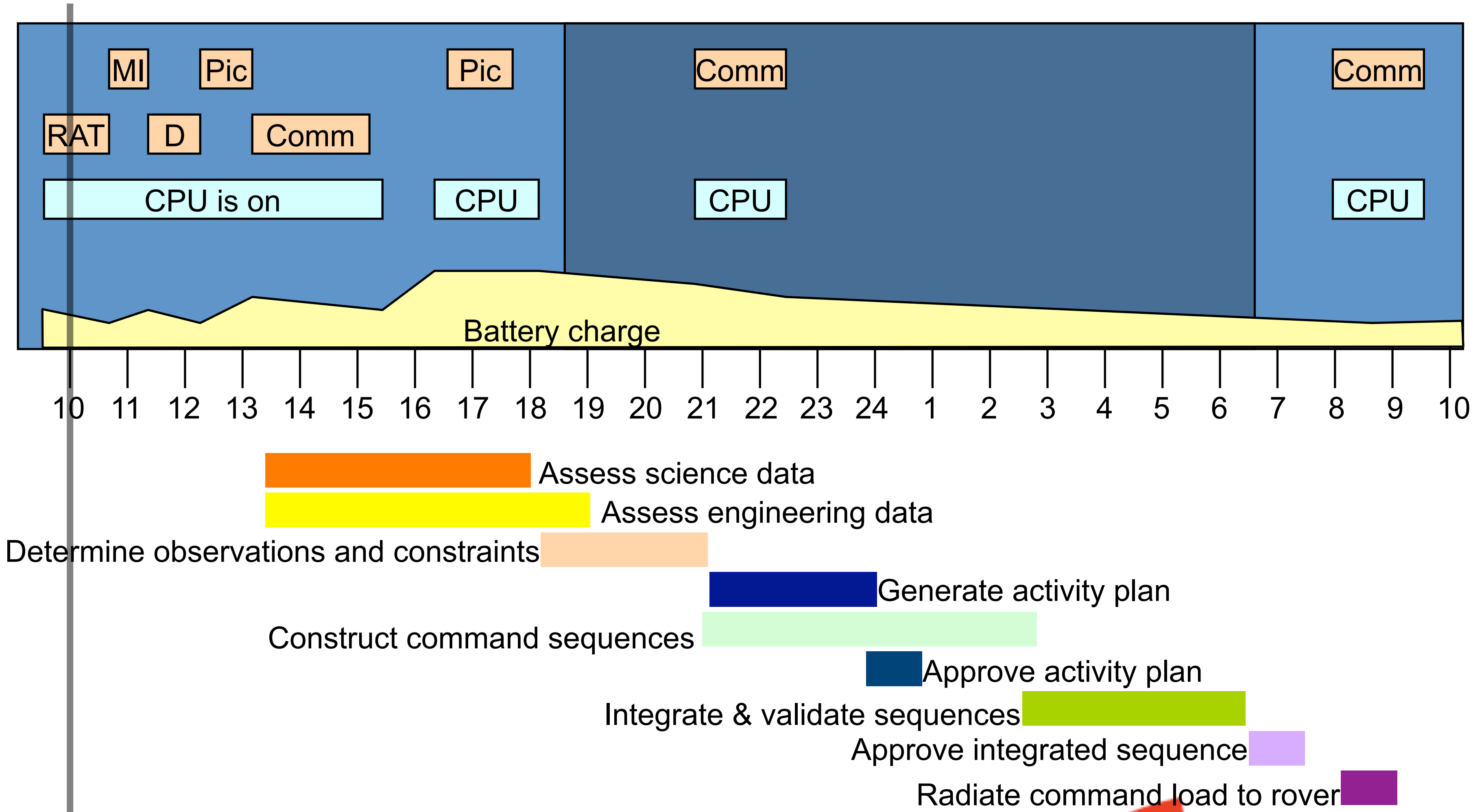
A day in the life of a MER rover

- Operations based on Mars time
  - Solar powered rover operations driven by day/night cycles
  - Planetary alignment and orbiter orbits drive communications
- Tight schedule for planning
  - Recent information from rovers required for plan generation
    - Terrain information needed for drives and arm movements
    - Scientists require up-to-date information about outcomes and results
  - Result: Minimize time from state information to plan start
    - About 19 hours available for overall process
    - But, only about 2 hours available for planning
- Caveat: What follows is not a real plan
  - Does not conform to preferred operations practices
  - Is much smaller than real plans which have up to a 100 high-level requests and up to 3500 low-level activities.

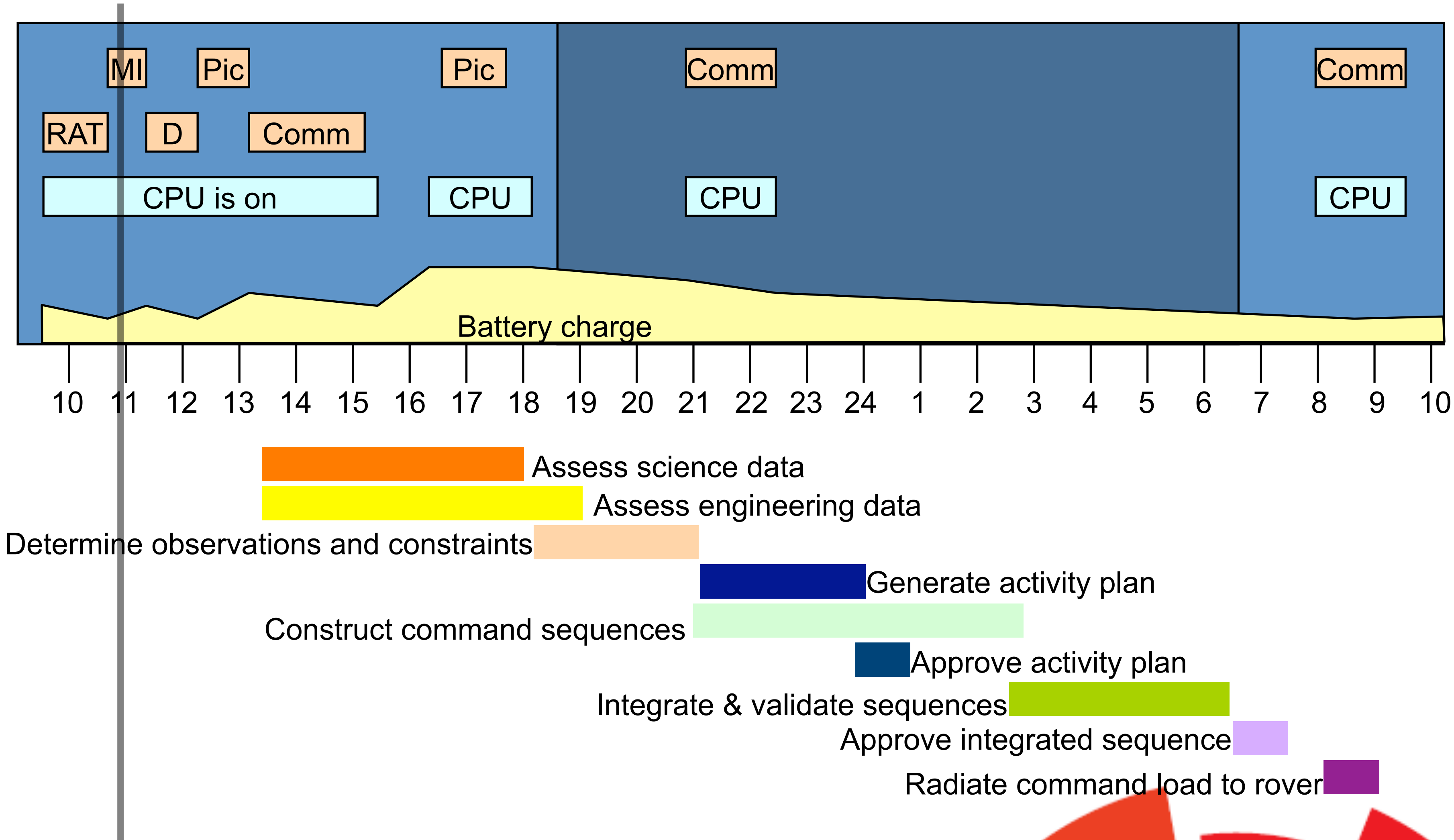
9:30 - A new plan for a new day



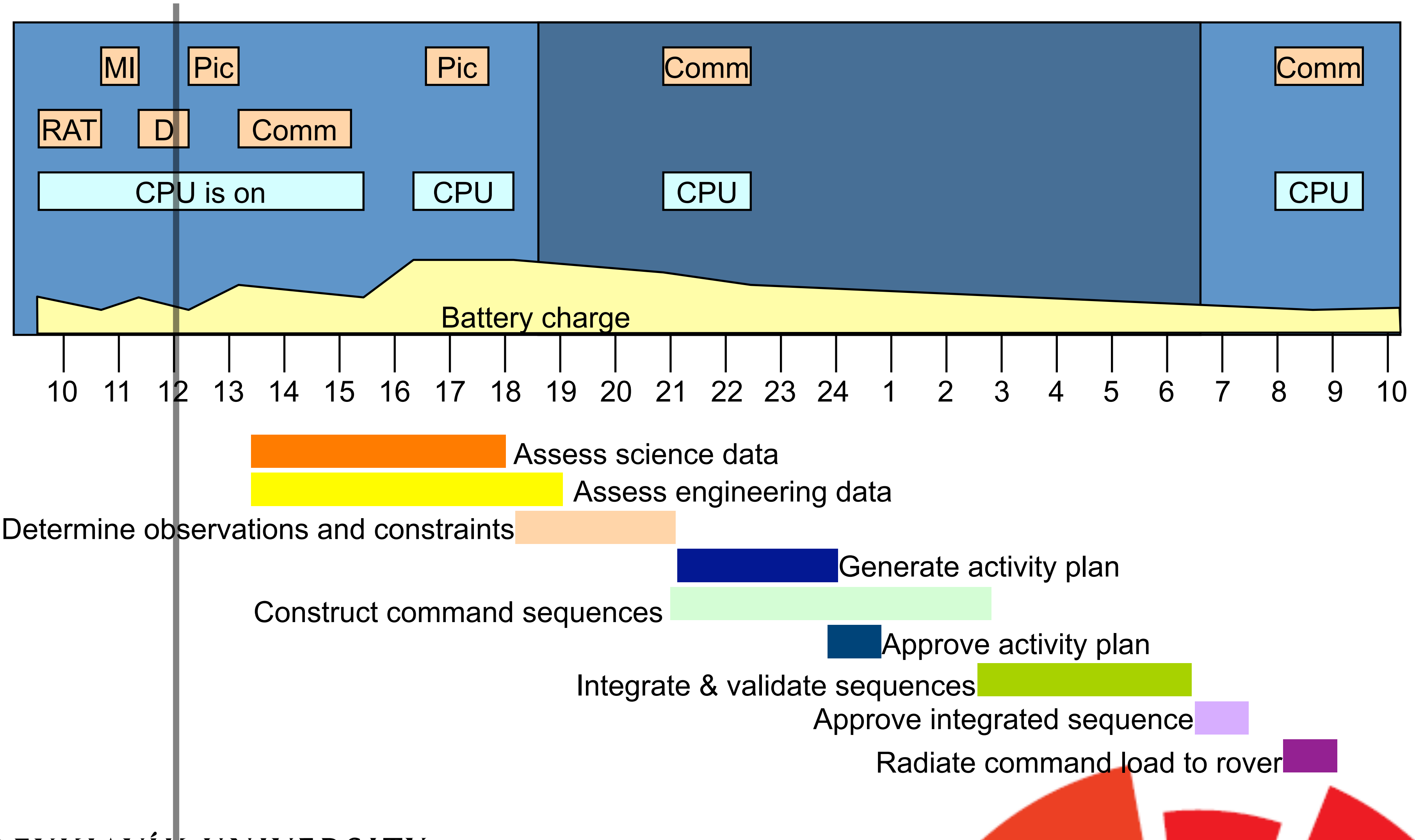
10:00 - Abrading a rock



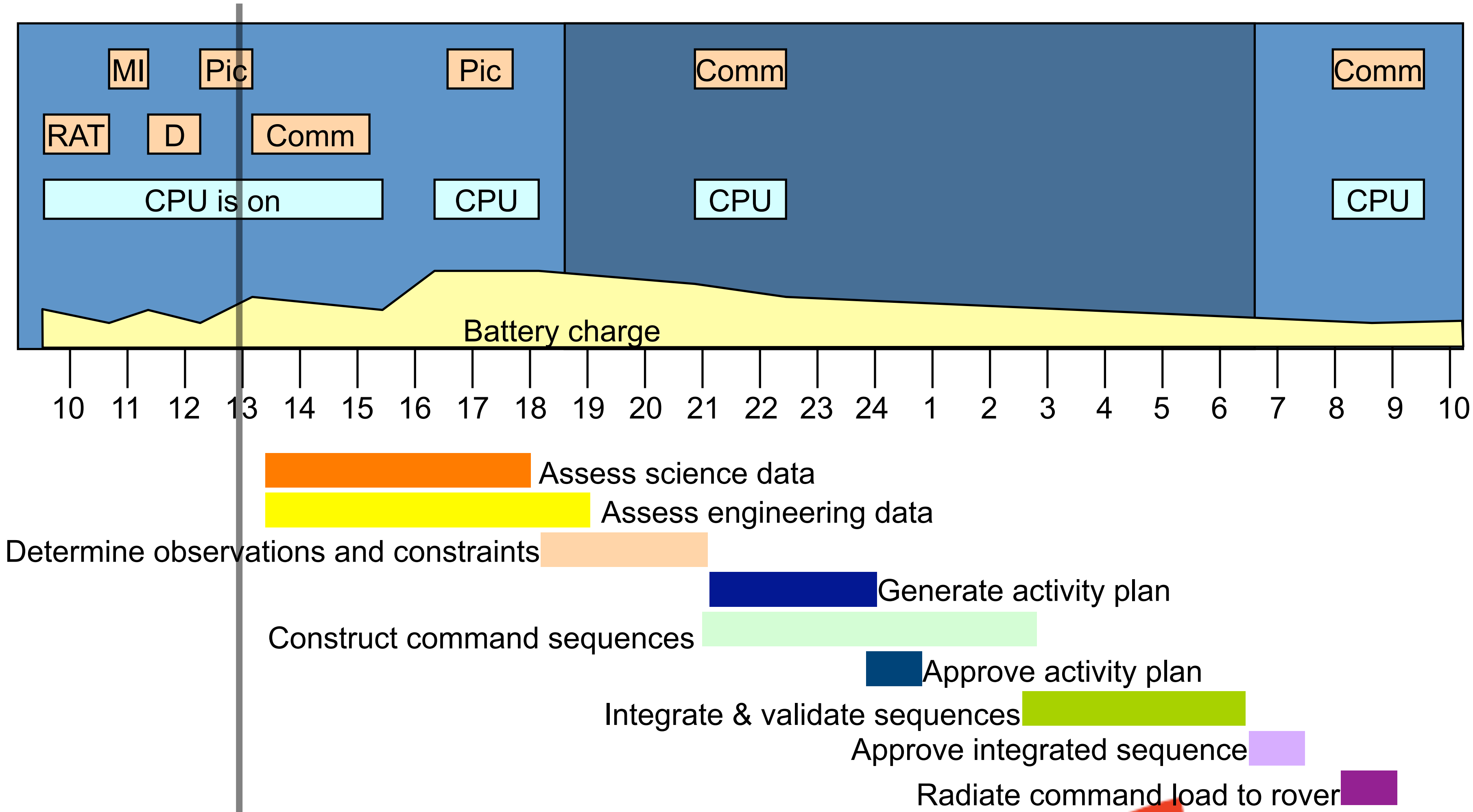
11:00 - Microscopic imaging of rock



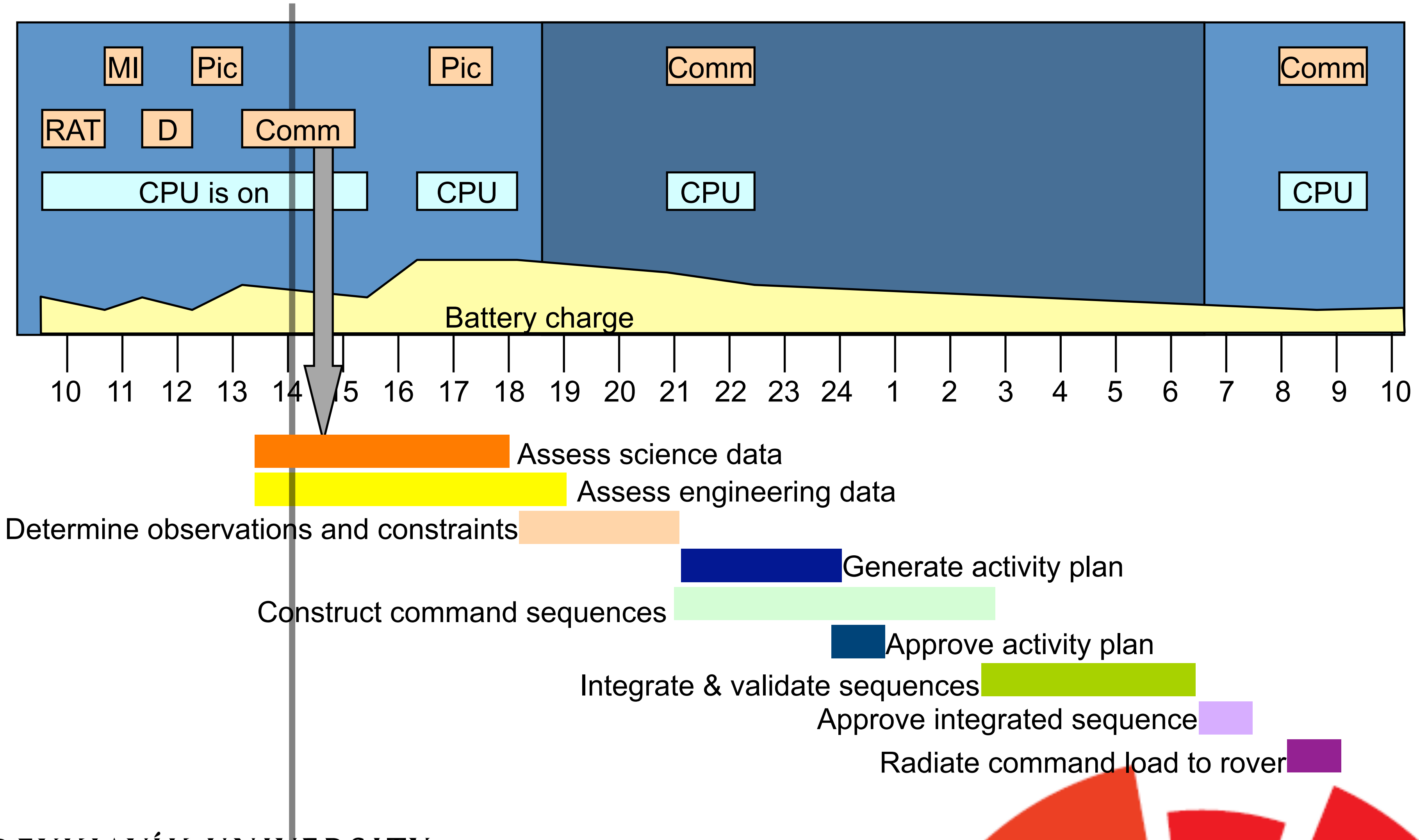
12:00 - Driving to a new location



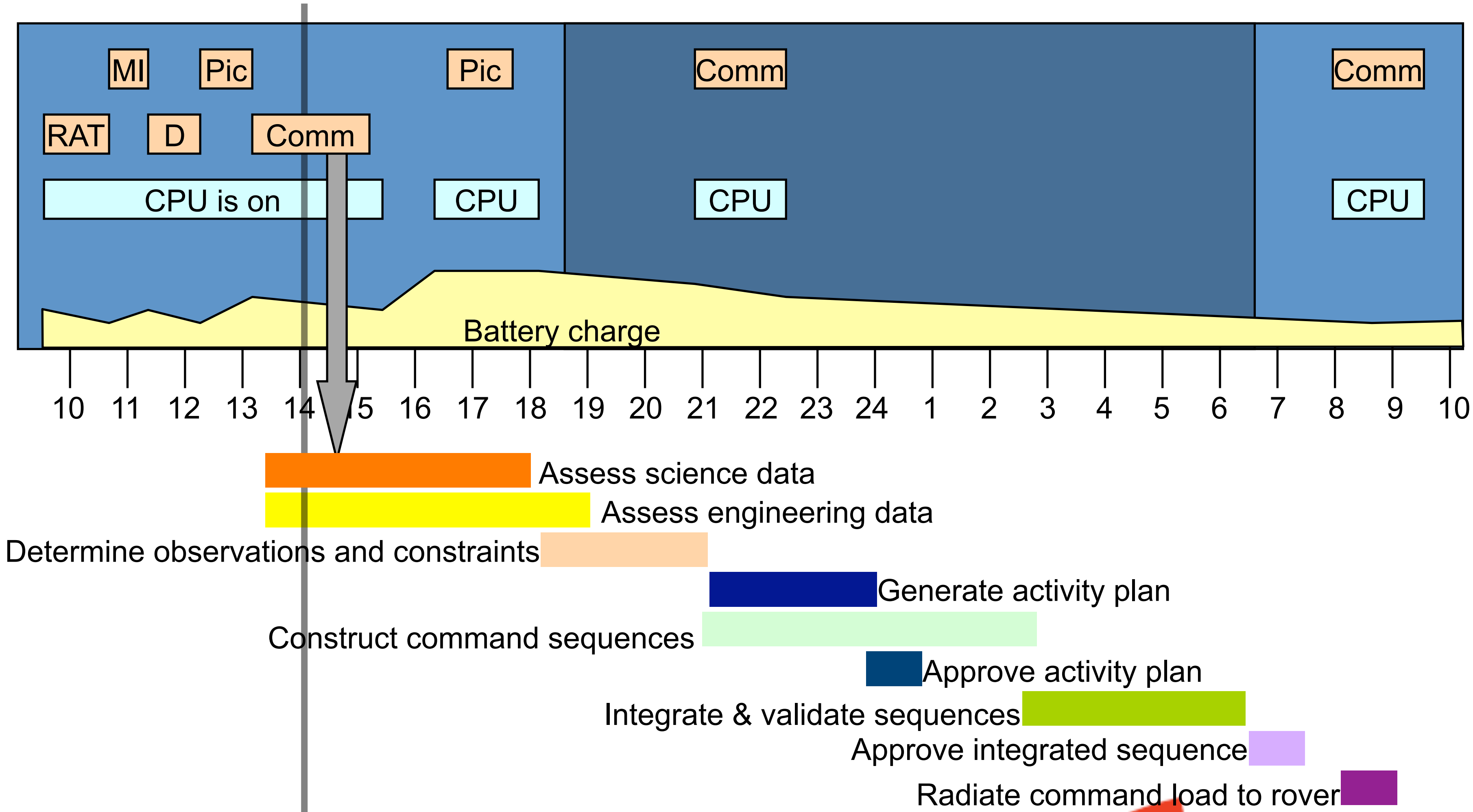
13:00 - Taking pictures of new location



14:00 - Sending data to Earth

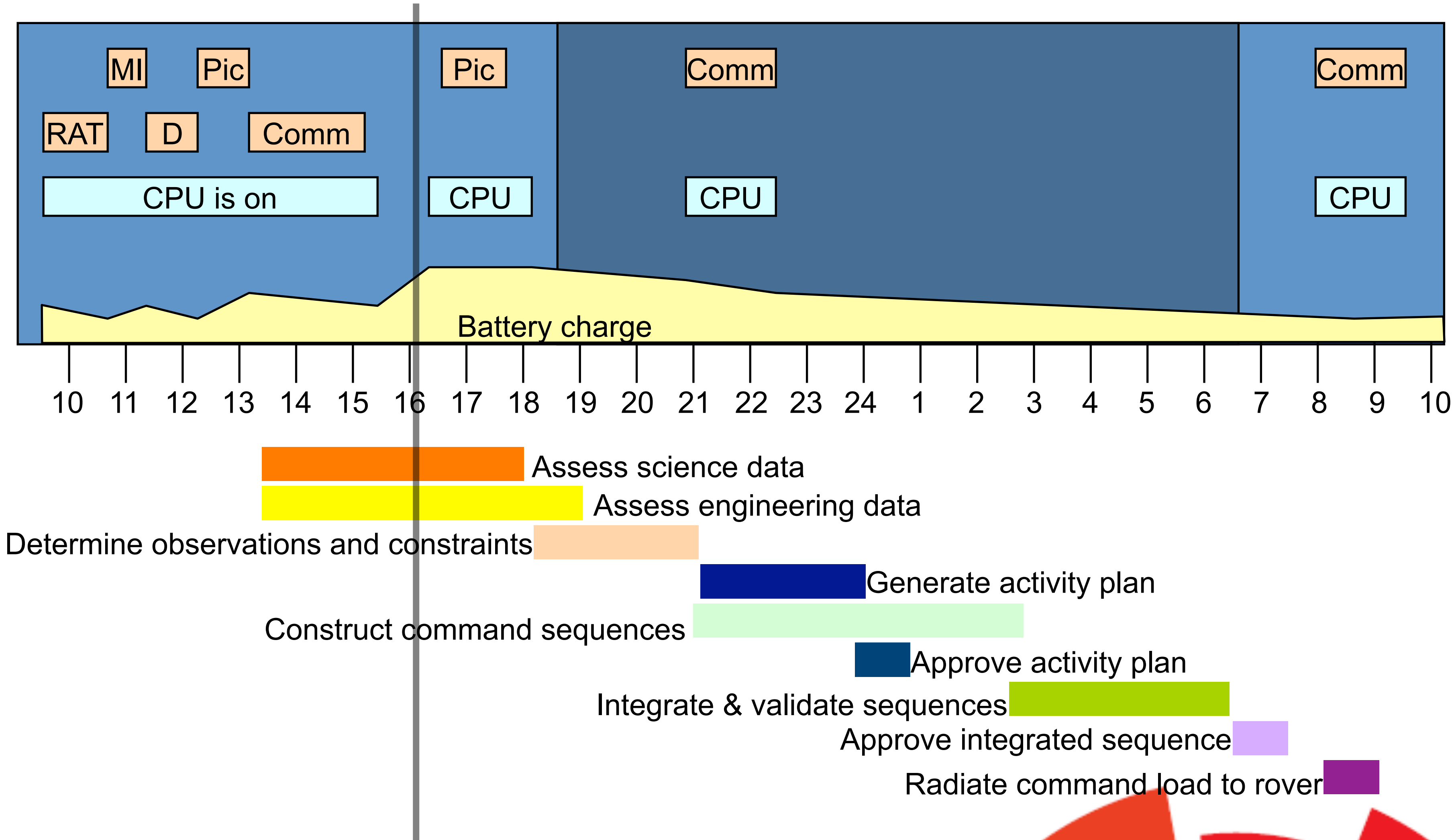


In the meantime, on Planet Earth…

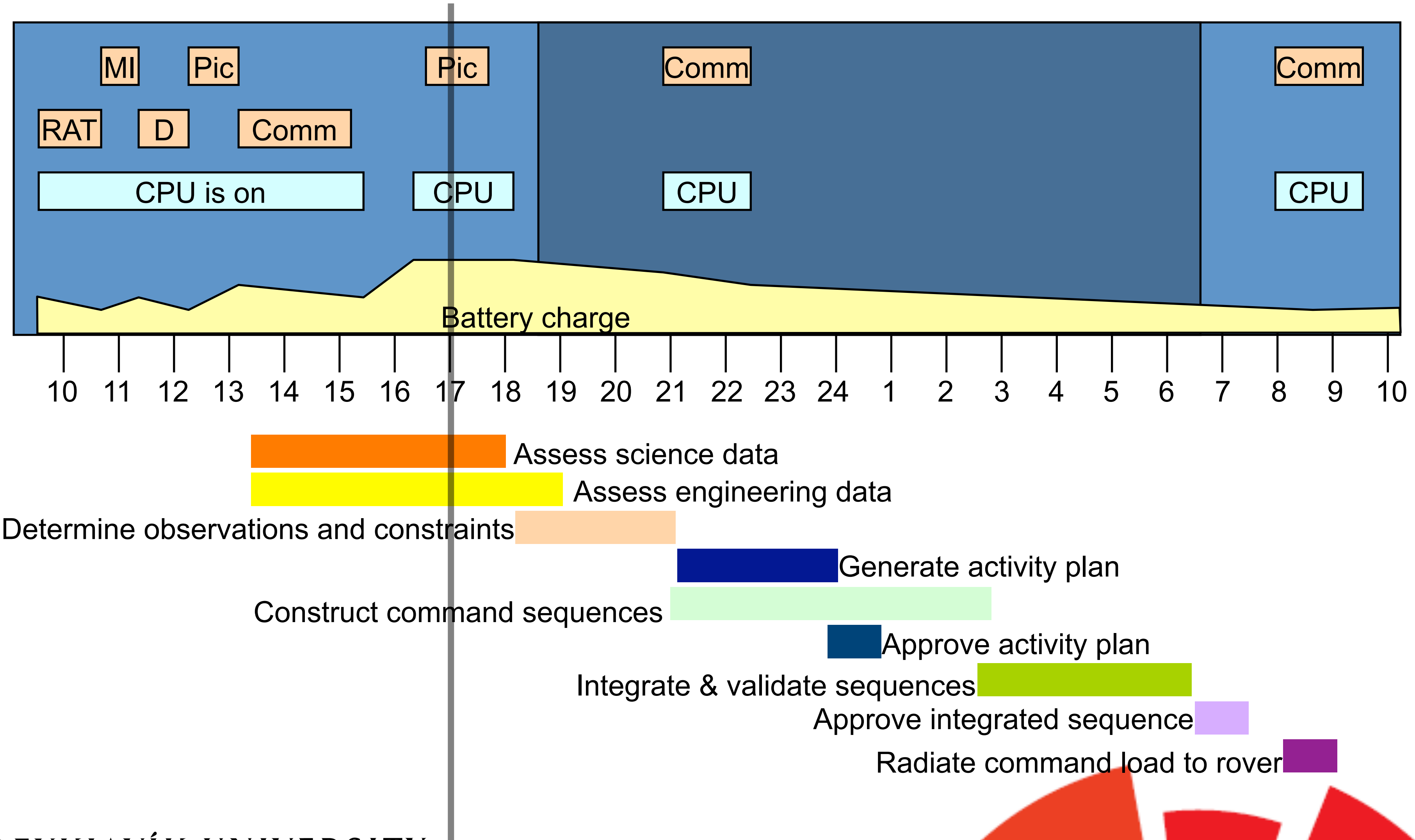


REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

16:00 - Taking a nap to recharge

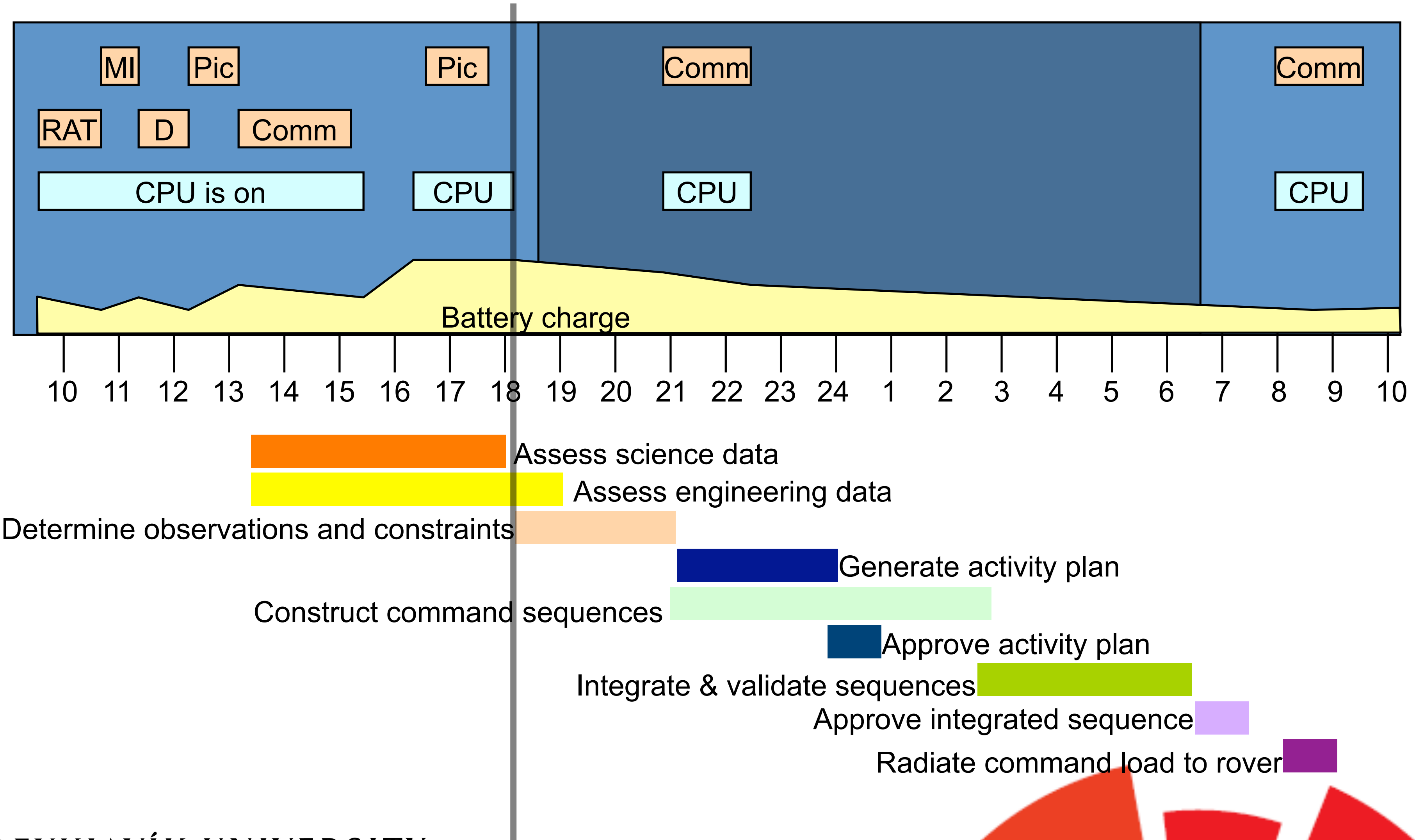


17:00 - Taking more pictures



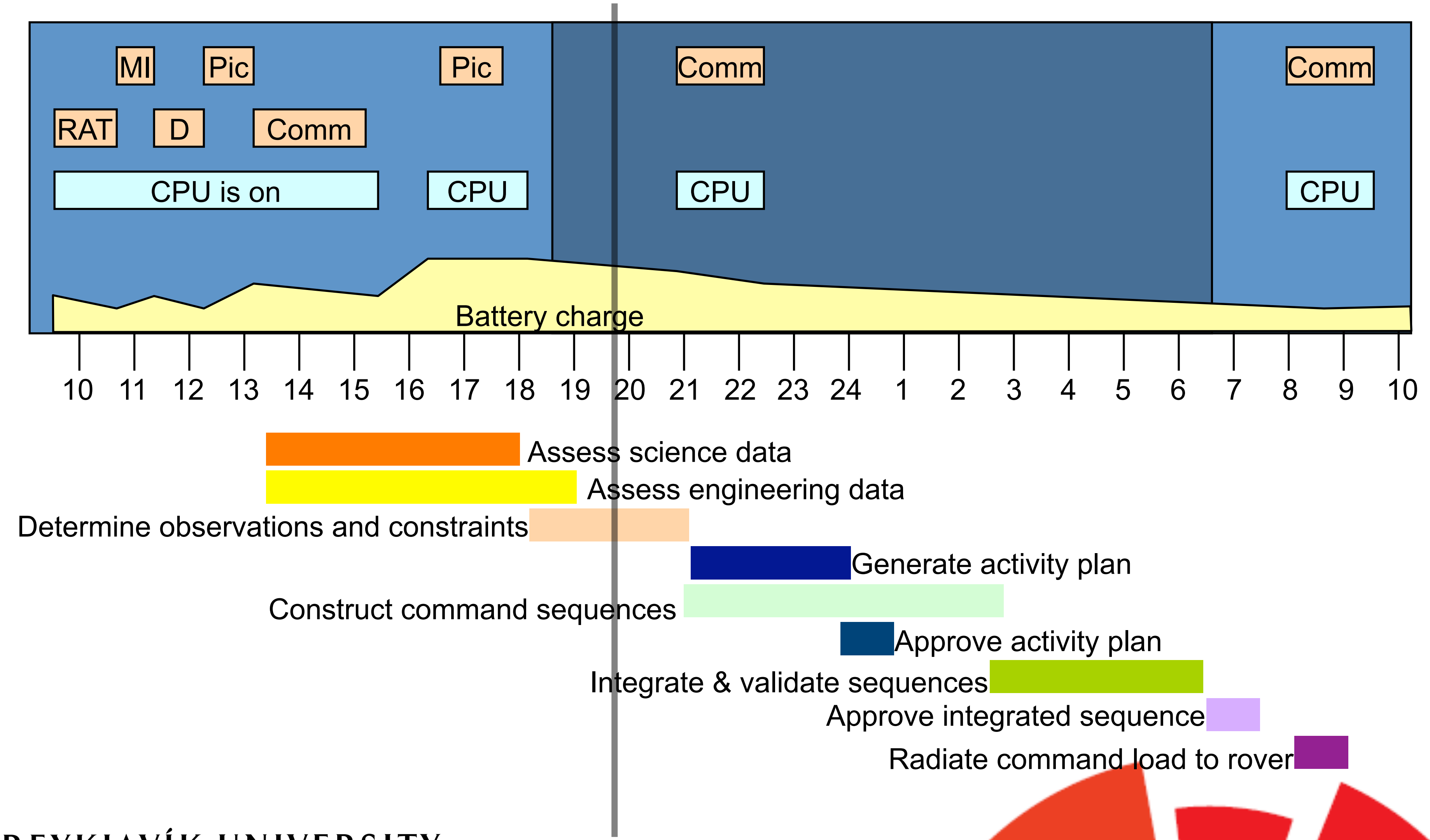
REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

18:00 - Going to sleep for the night

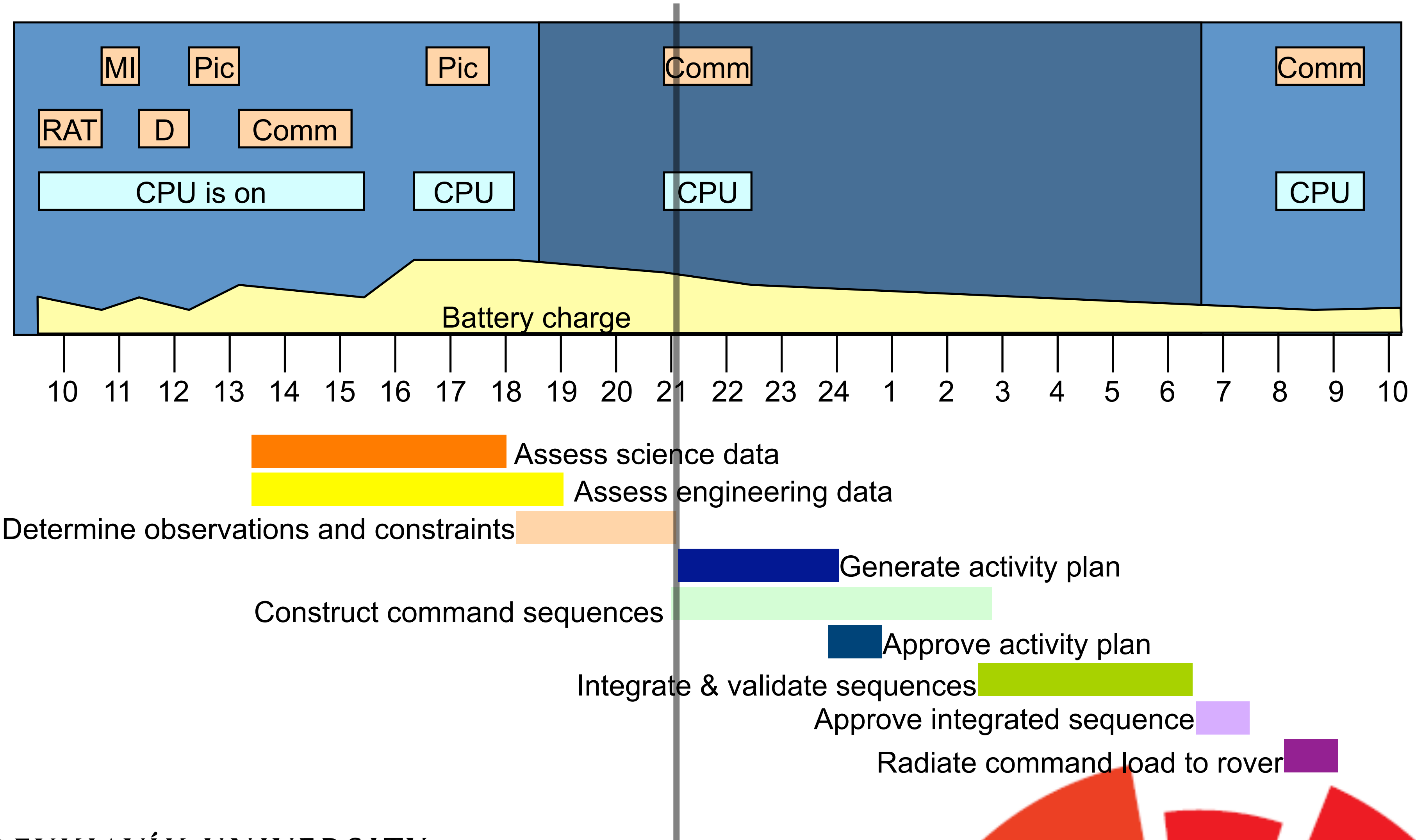


REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

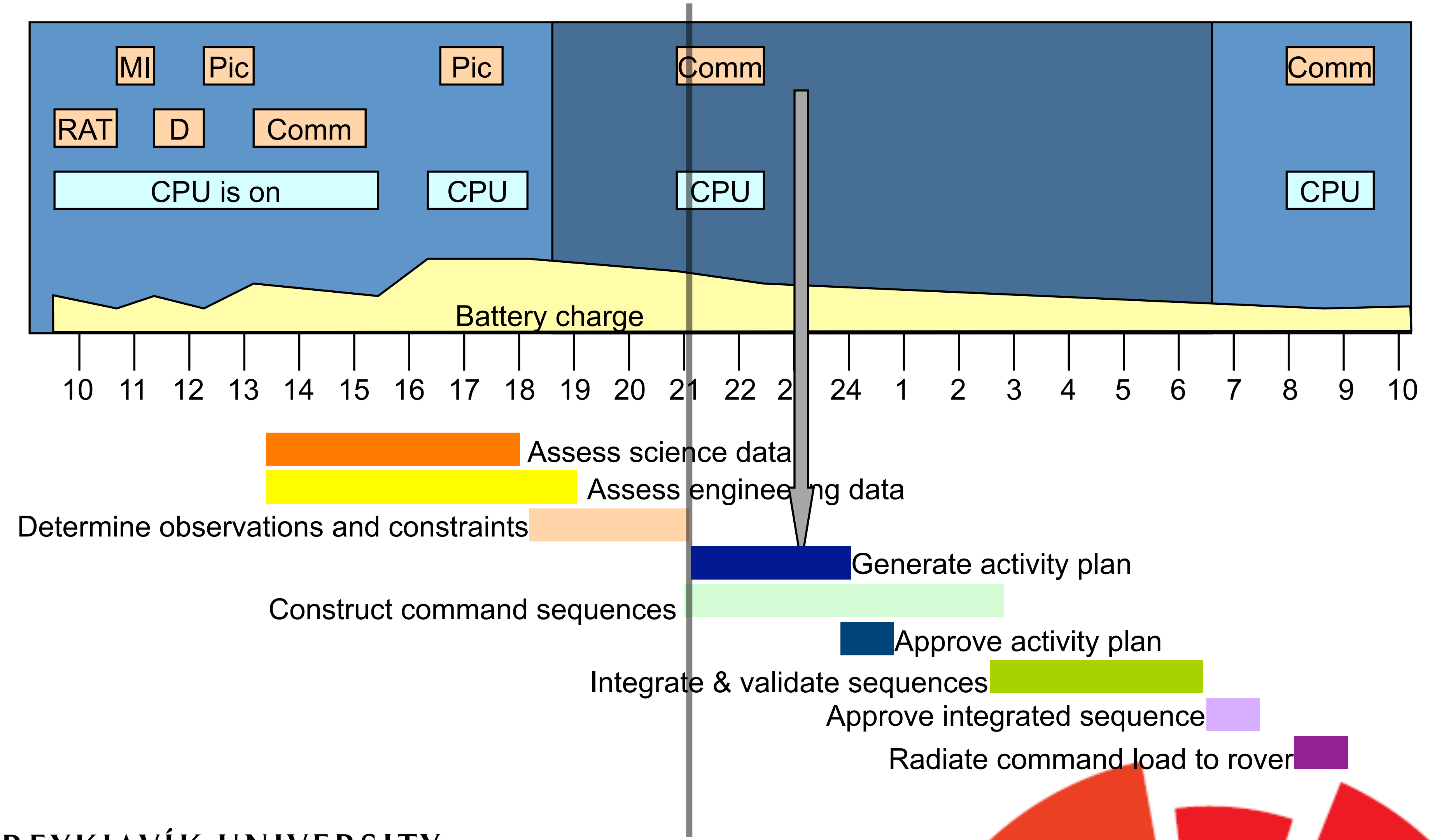
On Earth: Planning problem being defined…



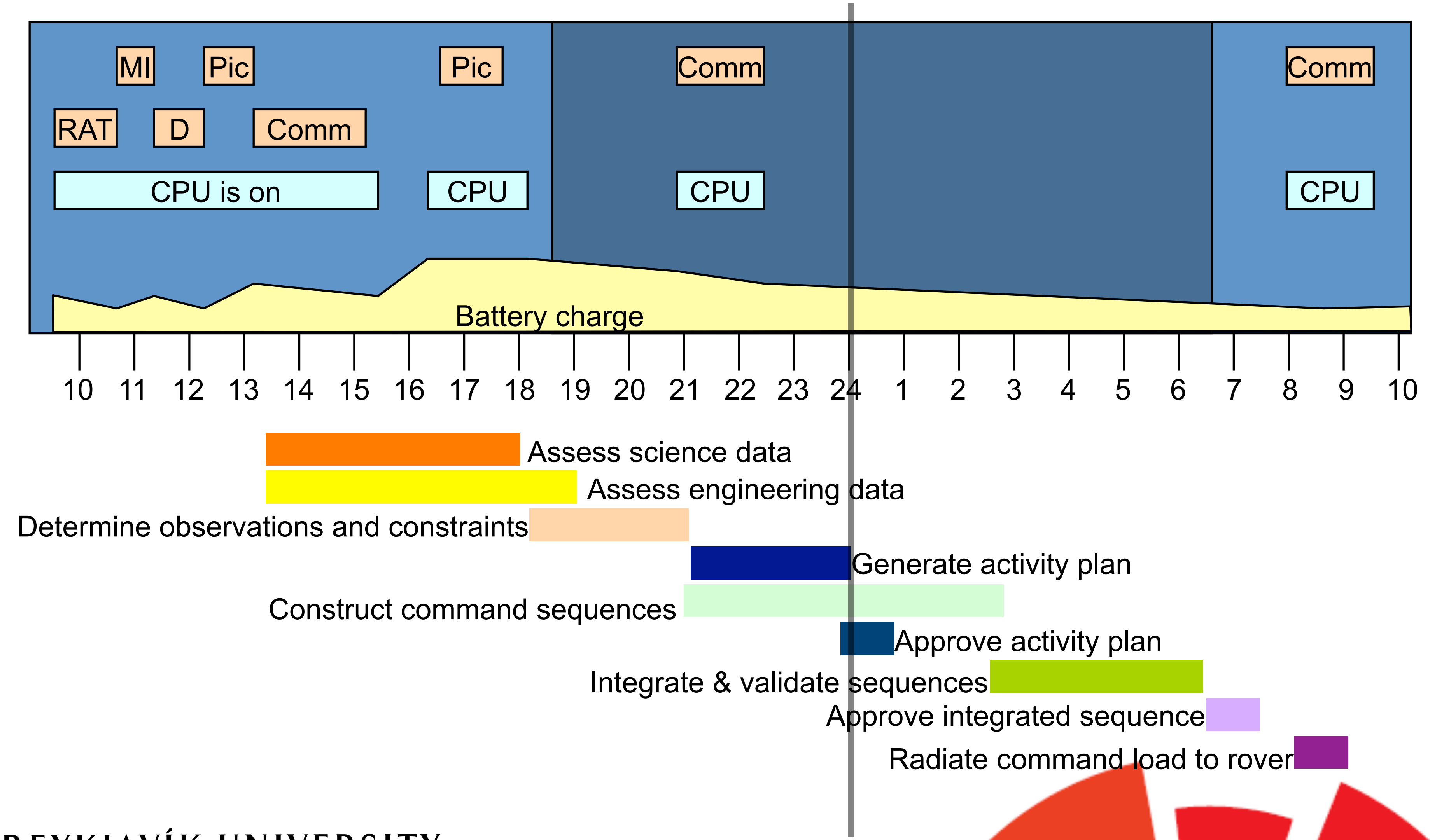
21:00 - Waking up to send data to orbiter



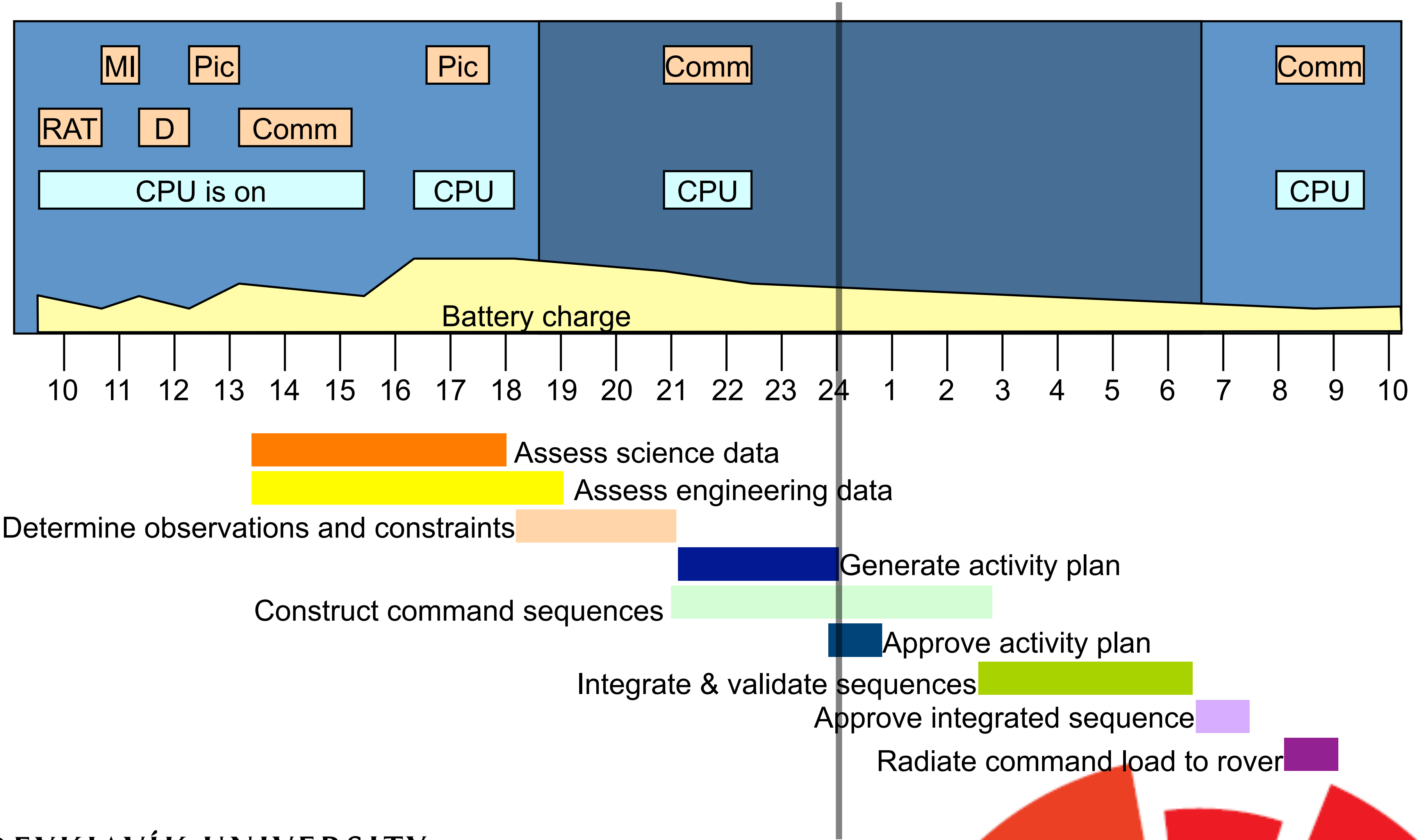
On Earth: Data may impact planning proces



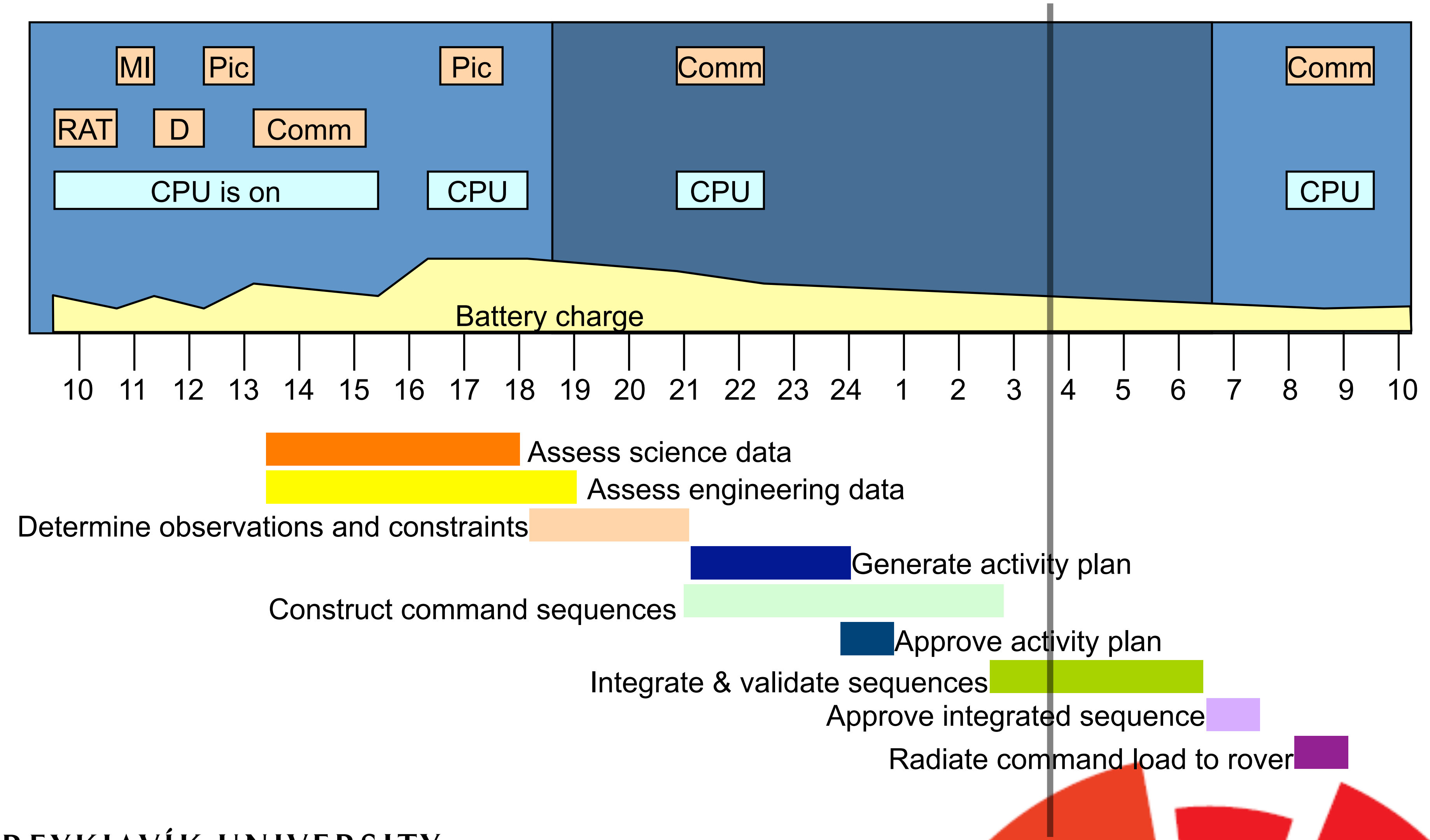
24:00 - Sleeping, but keeping warm



And on Earth: A plan is ready for approval

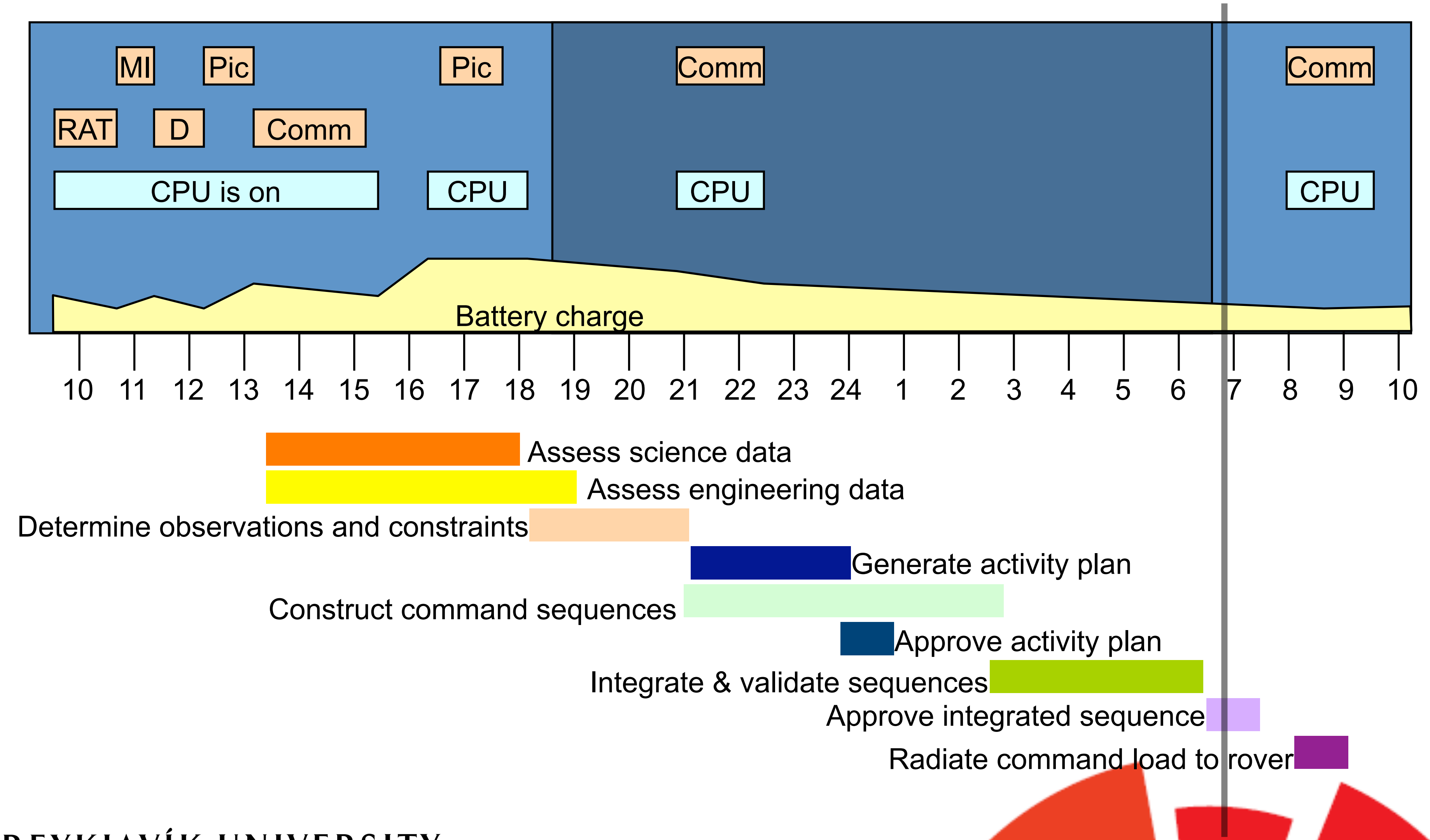


On Earth: Command sequences being built

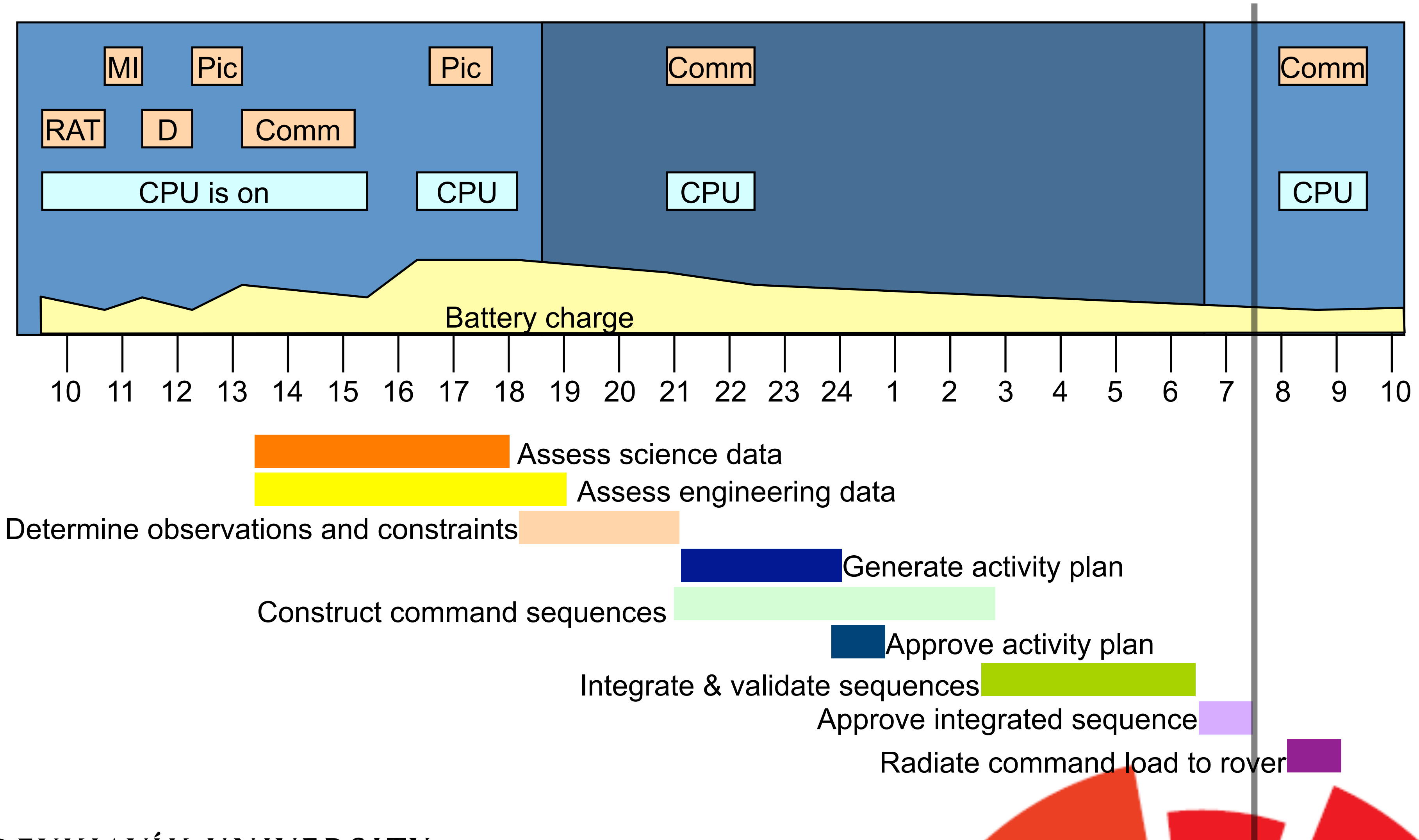


REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

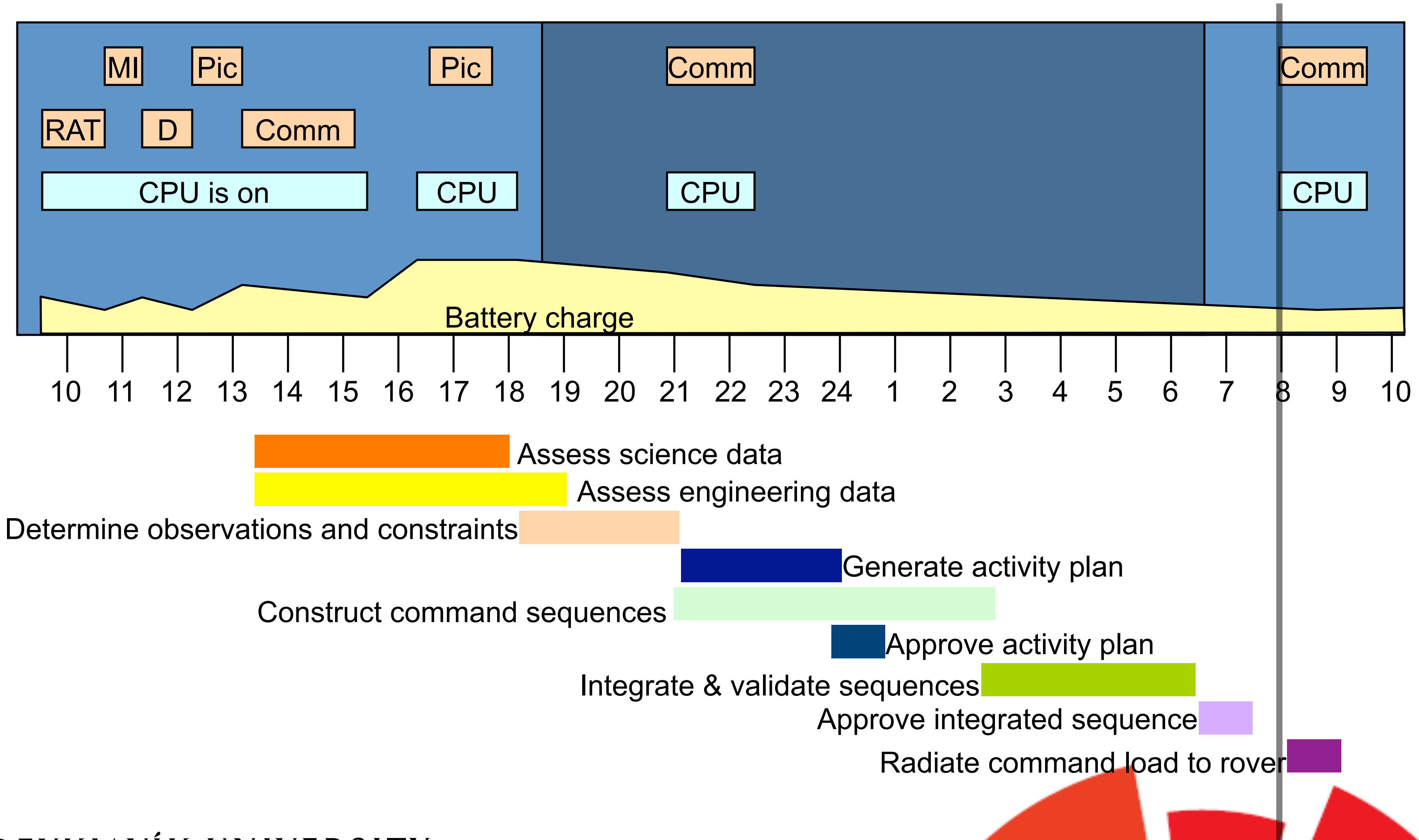
Approved


MI
Pic
Pic
Comm
Comm
RAT
D
Comm
CPU is on
CPU
CPU
CPU
Battery charge
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9 10
Assess science data
Assess engineering data
Determine observations and constraints
Generate activity plan
Construct command sequences
Approve activity plan
Integrate & validate sequences
Approve integrated sequence
Radiate command load to rover

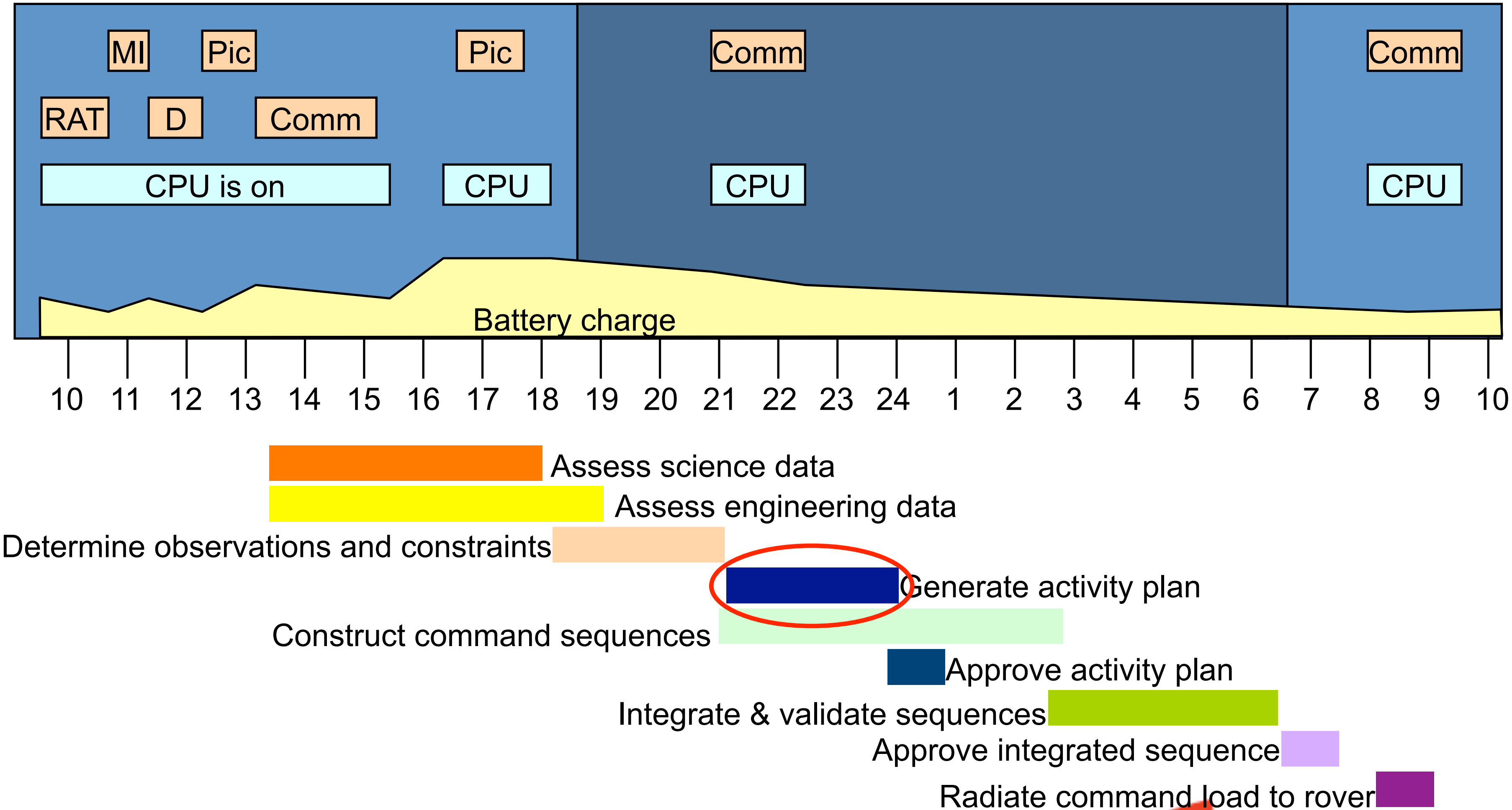
And ready for transmission



8:00 - Waking up to receive new plan



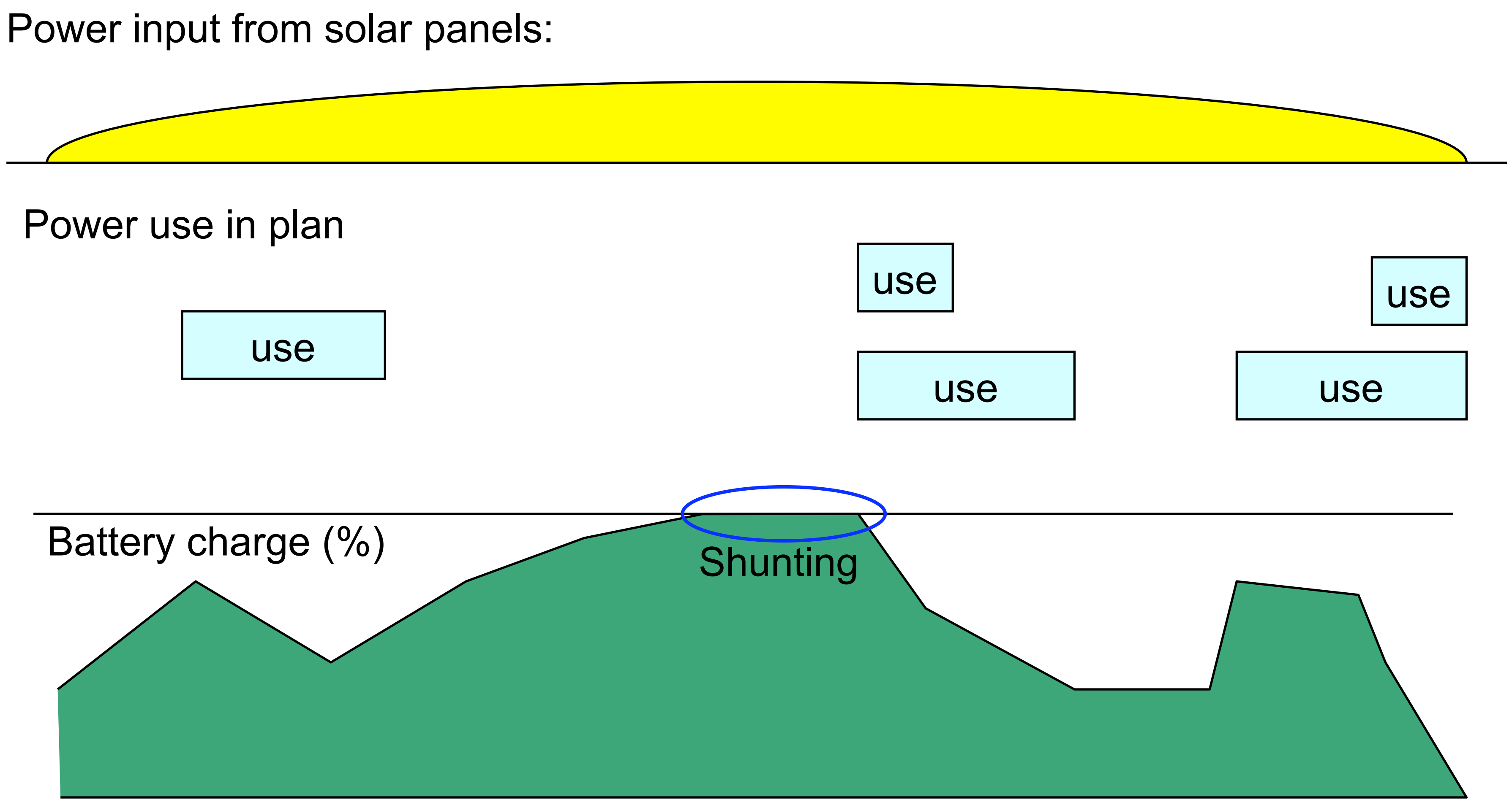
So, where is the planning done ?



Activity Planning Process

- Input
  - Prioritized requests from scientists and rover engineers
    - Temporally constrained
    - Temporally related
  - Predicted rover state at plan start time
- Output
  - A day-long plan for rover activities
    - Achieving as much of science requests as possible
    - Satisfying all specified constraints
    - Satisfying all general safety rules
- Rules
  - Decomposition rules for high-level activities
  - State rules: pre-conditions, maintenance conditions
  - Resource impact and resource limitation rules

Resource example: Battery charge



Complex Planning Problem!

- Constrained temporal planning with concurrent actions
  - Durative concurrent actions related by temporal constraints
  - State conditions, pre-conditions, effects, etc.
  - Exogenous events and constraints on action instances
- Oversubscription problem
  - More requests than can be fit
  - Goal is to maximize science, weighted by priorities
- Solution preferences
  - Among “equivalent” plans, some preferred over others
    - Temporal placements, natural ordering,…
- Complex resources
  - Nonlinear accumulation of impacts
  - External impacts on resource levels
  - Example: Battery charge on solar-powered rover

So, this is just a planning problem, right?

“PRACTICAL PLANNING”

Perfect for classical planning?

- Idea of classical planning in applications
  - Build the model and get plans automatically
  - With good heuristics, no problem, right?
- Problems
  - Any legal plan is not the desired solution
  - Sometimes the planning process is the key
  - Often the rules cannot be expressed
  - Planner application requirements change

Real world adds more complexity

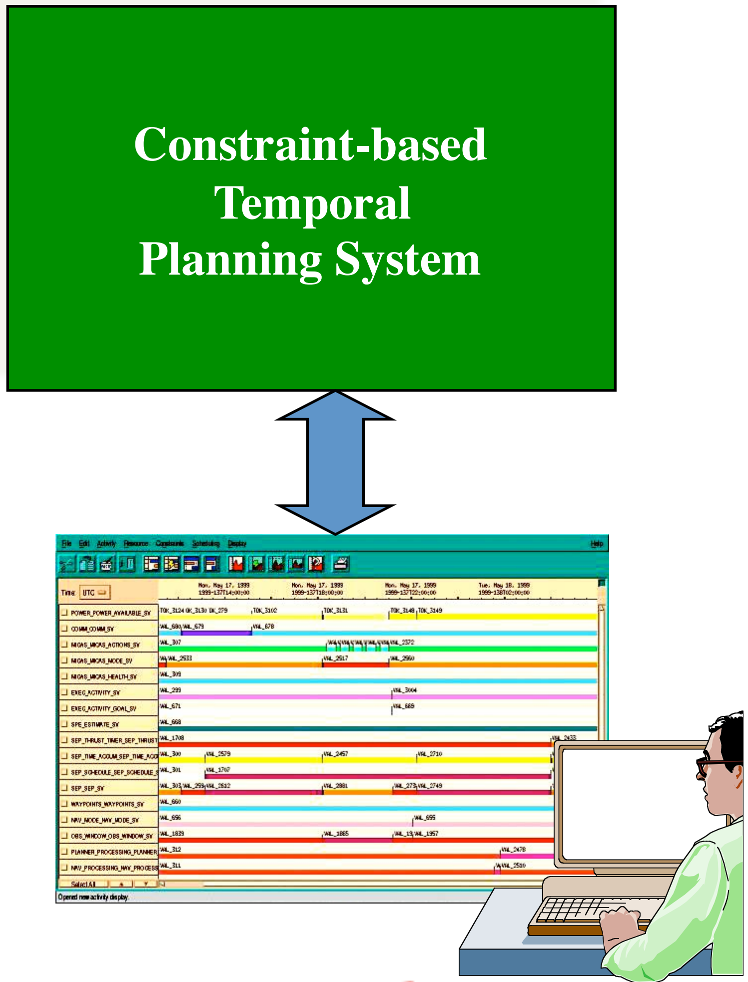
- Plan quality metric not fully specified
  - Quality metric had subjective aspects
  - Some preferences and quality metrics vary sol to sol
- Rules are not absolute
  - Rules are incomplete and exceptions can be made
  - Rover operations evolve and circumstances arise
- Rules and constraints very complex
  - Energy resource calculations and operational limits
  - Thermal protection rules
- Automated capabilities not desired for all cases
  - Resource leveling decisions handled by experts

So, what to do?

- Fully automated planning not possible
  - Not suitable for current rover operations process
    - Human evaluation of plans for safety
    - Require human understanding of plans
  - Subjective preferences play a significant role
    - Cannot be captured in current domain definition languages
  - Some rules exceed domain modeling capabilities
    - Power and thermal levels calculated using finite-element analysis that is only applicable to fully instantiated plans and takes 5 minutes to run
- Need to permit human involvement
  - Humans can make subjective evaluations
  - Humans-in-the-loop facilitate understanding of plans

Solution: Mixed-initiative planning

- MAPGEN
  - Back-end: Constraint-based planning system called EUROPA
  - Front-end: Existing mission plan editing tool called APGEN
- Core notions:
  - Actively enforce constraints as user edits the plan
  - Offer useful operations for large-scale plan editing

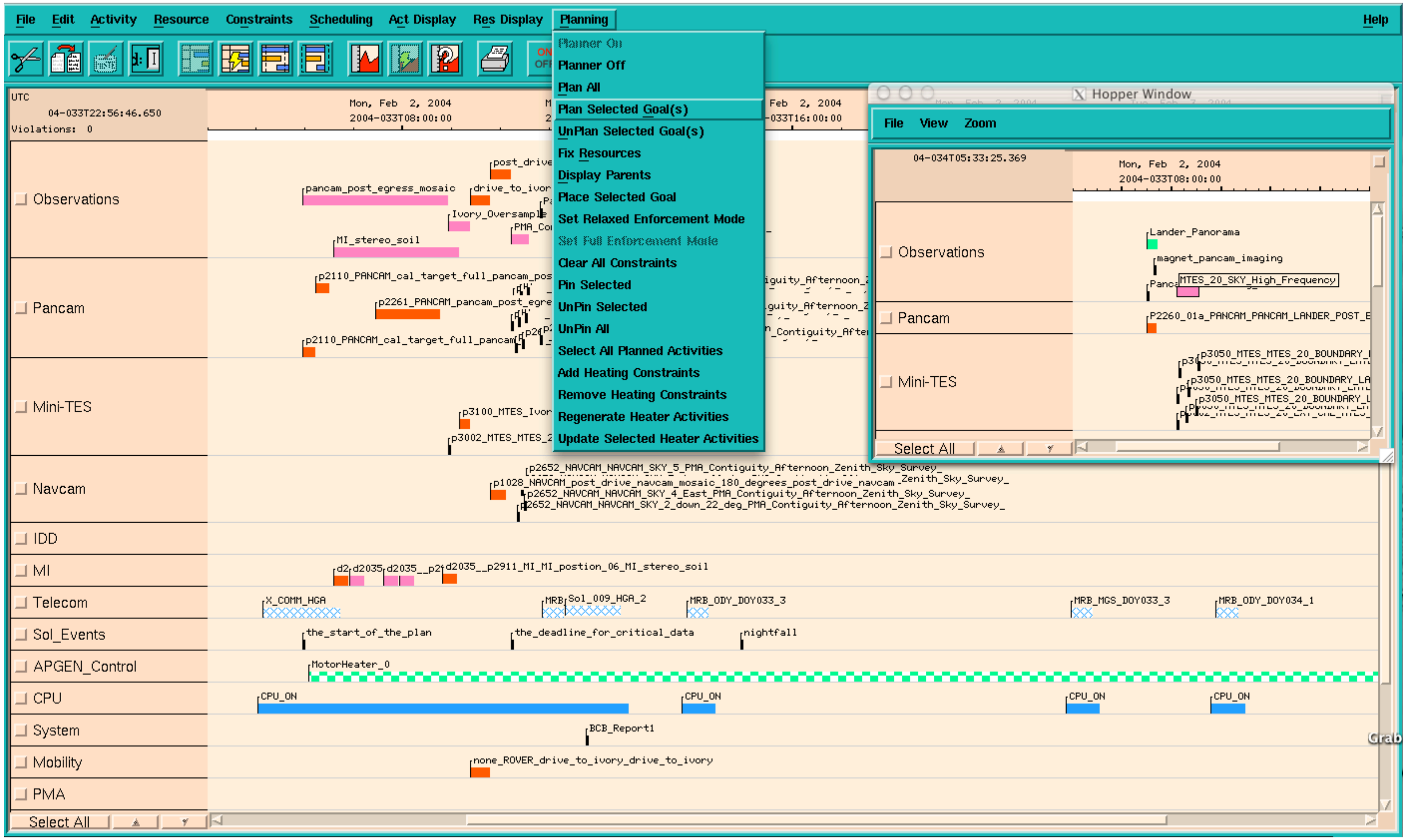


User operations

- Scheduling requests
  - Add pending requests to the plan
  - Range from one activity at a time to fully automated planning
- Unscheduling requests
  - Move activities out of the plan, making them pending requests
- Plan editing
  - Adding, deleting and modifying activities
- Moving activities
  - Move activities in time

- Always: Maintain validity of solution

MAPGEN interface



Usage scenario

- Repeat until plan is ready:
  - Add more requests to the plan
    - Different users used different approaches
  - Evaluate plan
    - Resource usage
    - Subjective preferences
  - Modify plan as needed to
    - Satisfy resource limitations
    - Satisfy rules that are not in domain model
    - Respect preferences to the extent possible

- In case of new information about rover:
  - Make necessary changes to plan

Development of MAPGEN

- Development timeframe short
  - 2001-2003 (for whole mission)
- Good lesson about applying planning
  - Flexibility and adaptability keys
- Unclear ground software architecture
  - Changed repeatedly and often significantly
- Late issue with user acceptance
  - Led to new approach with time display

Abridged history of MAPGEN

- Call for help with mission in late 2000
  - Initially, offer of automated planning rejected
- Prototypes and demonstrations in 2001
  - Results in acceptance into mission
- Initial development and testing “in vacuum” in 2002
  - Many missteps and problems
- Major descope in ground software in 2002
  - Major change in planned use of MAPGEN (relation to SAP)
- Another descope for MAPGEN in 2003
  - Results in “Constraint Editor” tool among other things
- Operations tests in 2003 identify new changes
  - Problems with planner behavior
  - Asked to also handle heaters
- MAPGEN accepted and used shortly after landings in 2004

Main deliveries and tests

- 7.1 - delivered in Dec 2002
  - Speed improvements
  - Goal rejection
  - Parent-child activity tracking
  - Importing APGEN expansions
- 7.6 - delivered in March 2003
  - Resource leveling,
- 8.5 - delivered in July 2003
  - Super-move
  - User preferred times
  - Relaxed mode
- 9.1 - delivered in ?????? -
  - Planner restore
  - Sol events
  - Auto-completion on read-in
  - Redetail handling
  - Group/ungroup suppport
- 10.1 - Operations redelivery in March
  - Three minor planner bugs fixed
  - Major APGEN editor error fixed

- R1 - June 2001
- R3 - Sept 2001
  - Demonstration to mission
- TTC - July 2002
  - Performance abysmal
- TTE - Dec 2002
  - Initially limited to acceptance test
  - Unexpected participation
- TTG - April 2003
  - First hands-on use by TAPs
  - Some crashes and "turn-offs"
  - Some unexpected use
- PORT 3 - August 2003
  - Very limited use in actual tests
    - Problems with constraint editor use
    - Some instability
    - Problems with recovering plans
- "Bake off" thread tests
  - Preceded by comprehensive training
  - Sequence of tests by various users
- PORT 6 - Nov 2003
  - Significant improvements over PORT 3
- Surface Operations - Jan 2004
  - Few hard crashes
  - Some annoying workarounds

Would classical planning have worked?

- Primary issues
  - Plan generation was not a key target
  - Application way beyond expressivity of classical planning at that point
- Secondary issues
  - General heuristics would not have worked
  - Integration into editor application tricky
  - Time-bounded optimization required
  - Plan editing interleaved with planning

Still, general planning is important

- General planning key to success
  - Responsiveness to changes
  - Ability to handle unforeseen changes
- Planning applications very complex
  - Too expensive to do algorithms from scratch
  - Validation is a critical issue
- Exact planning approach not main issue
  - Advances often transfer between approaches

So, what did we use?

CONSTRAINT-BASED PLANNING

Constraint-based planning

- Family of approaches based on:
  - Time as network of variables
  - Timelines describing fluents over time
  - States and activities networks of variables
  - Constraint propagation
  - Expressiveness in goals, external events etc.
- Will use EUROPA as example
  - Probably most used in applications, along with ASPEN (see later)

Others than EUROPA

- Many other constraint-based planners
  - ASPEN (JPL)
  - IxTeT (LAAS)
  - CSP (ISTC/CNR)
- Also have various applications
  - EO1 on-board science planning
  - MEXAR download operations planning
  - Deep Space Network scheduling
  - and many more

Example of “Practical Planning” technology

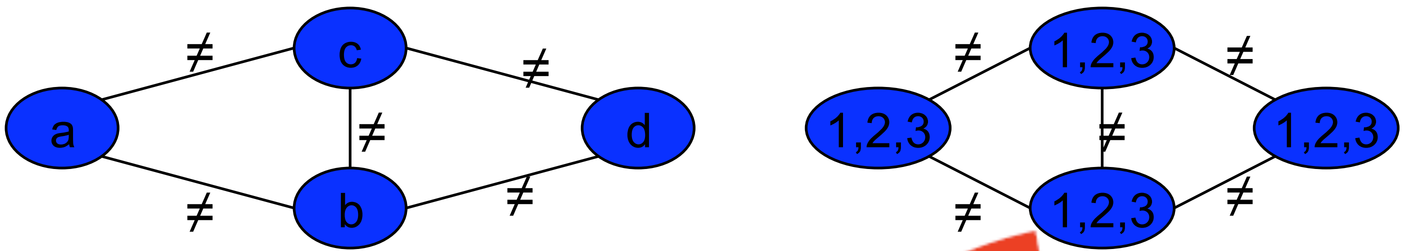
ABOUT EUROPA

Motivation

- Requirements driven by domain needs:
  - Concurrent operations with temporal dependencies
    - Instruments, mobility, heaters, communications, etc.
  - Limited resource availability
    - Power, data storage, equipment, etc.
  - Complex rules for interactions between operations
    - Example: Instruments require heating, interact with communications and mobility operations
- Additional considerations:
  - Efficiency and power of constraint reasoning
    - In particular: Temporal reasoning and activity scheduling
  - Flexibility in plan completeness criteria and generated plans
  - Applicability of systematic methods

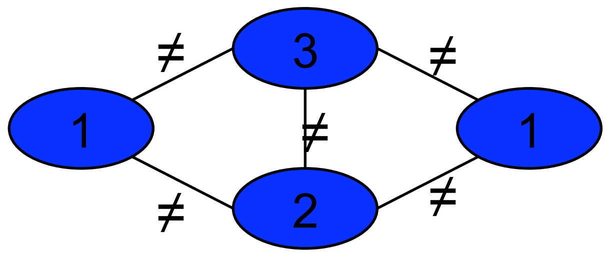
Constraint problems

- Constraint satisfaction problem
  – Set of variables, each with a finite domain
  – Set of constraints, restricting combinations of values
- Solution to constraint problem
  – Each variable assigned value from its domain
  – All constraints are satisfied
- Simple example
  – Variables: a,b,c,d take values from domain {1,2,3}
  – Constraints:

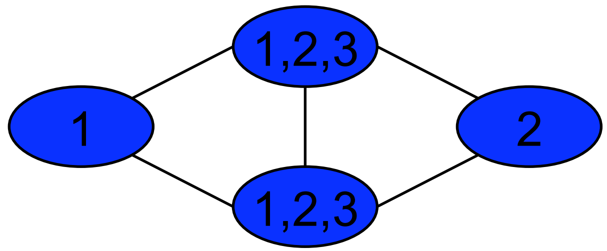


Constraint reasoning

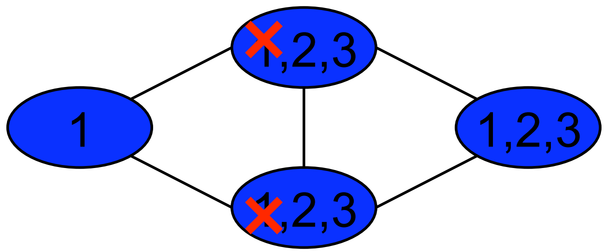
- Find solution:
  - Find values satisfying constraints



- Determine consistency:
  - Problem is consistent if a solution exists, inconsistent otherwise

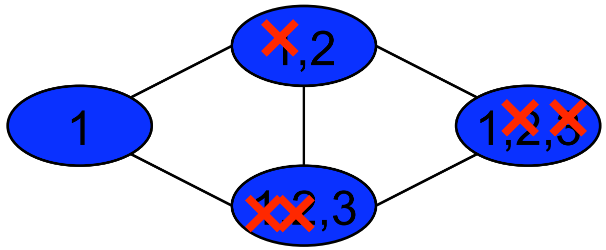


- Eliminate impossible values:
  - Value is eliminated if it cannot appear in any solution



Arc consistency

- Binary constraint is arc-consistent if
  - for every value in one variable
    there exist satisfying value for other variable
- Problem state is arc-consistent if
  - each constraint is arc-consistent
- Achieving arc-consistency
  - eliminate values for which no matching satisfying value exists
  - repeat to quiescence
- If a domain becomes empty
  - problem state is inconsistent
- If no domain is empty
  - problem state may or may not be consistent



Dynamic constraint problems

- Constraint problems as part of larger problem
  - Constraint-based planning
  - Design synthesis
  - Automated diagnosis

- Constraint problems change over time
  - Variables and constraints are added and deleted
  - Elements of domains are added and deleted

- Dynamic constraint satisfaction problems

Simple temporal reasoning

- Temporal constraint network
  - Variables represent event times
  - Constraints relate event times
- Simple temporal network
  - Domain of each variable is a temporal interval
  - Constraints specify distance bounds on variable pairs
- Efficient reasoning for simple temporal networks
  - Consistency can be determined in polynomial time

General constraint reasoning approach

- Dynamic constraint reasoning using procedures
  - Procedures replace (some) declarative constraints
  - Emphasis on propagation, not search (planner does search)
  - Permit variables and values to be dynamic
  - Support real-valued reasoning (within limits)

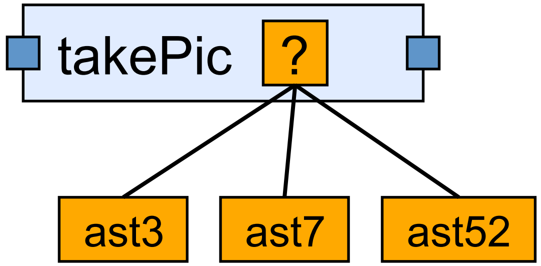
- Provide a general, widely applicable framework:
  - Other real-world planning systems
  - Configuration systems
  - Design synthesis
  - Automated diagnosis

Constraint reasoning for planning

- Representation
  - Represent activity parameters and temporal events
  - Represent constraints among parameters/events
- Reasoning
  - Identify when plan candidate is inconsistent
  - Eliminate choices not leading to valid plans
- Requirements
  - General: arbitrary constraints (domain-dependent)
  - Dynamic: constraints, variables and values added/deleted
  - Efficient: network changed and queried at each plan step
    - Trade-off between efficiency and completeness of reasoning

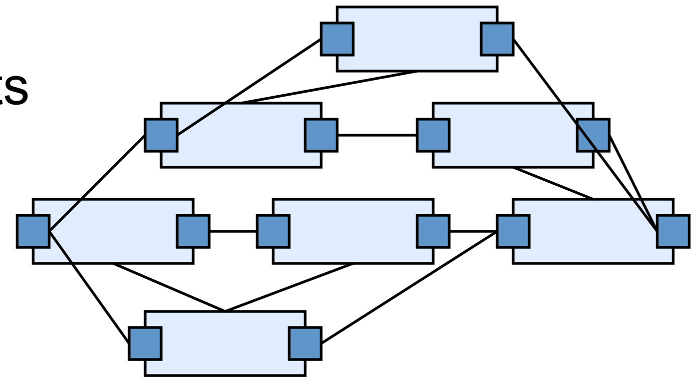
Constraint-based planning

- Activities represented as intervals
  - Each interval specifies activity
  - Each interval has start and end
  - Interval can have parameters



Candidate plan is a network of intervals

- Intervals linked by temporal constraints
- Interval parameters linked by constraints
- Gives rise to constraint network

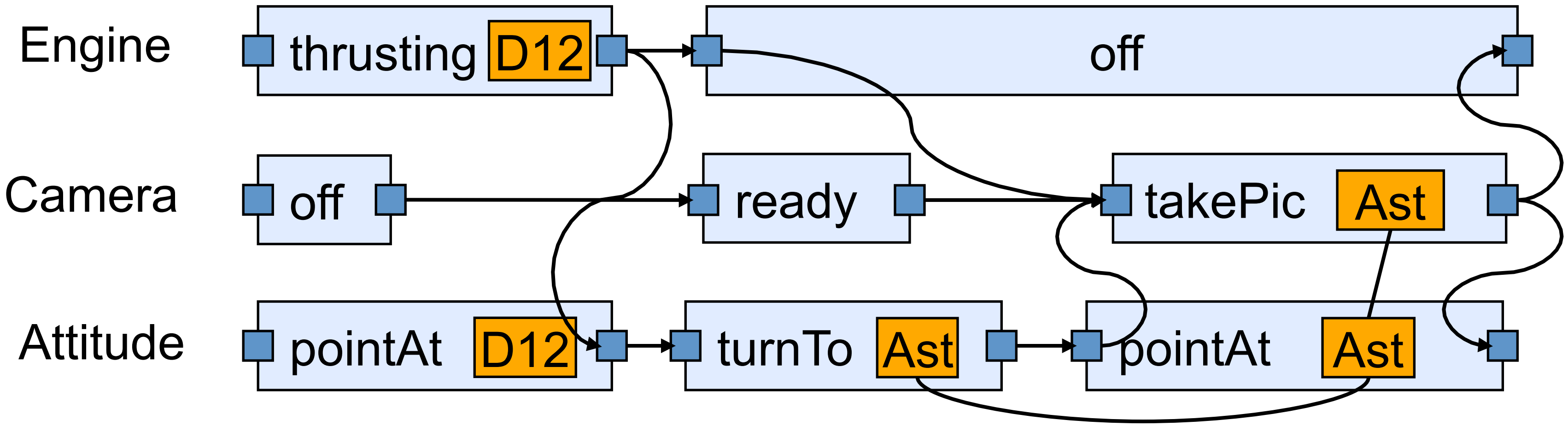


Feasibility of candidate plan

- If network is inconsistent, cannot become a valid plan

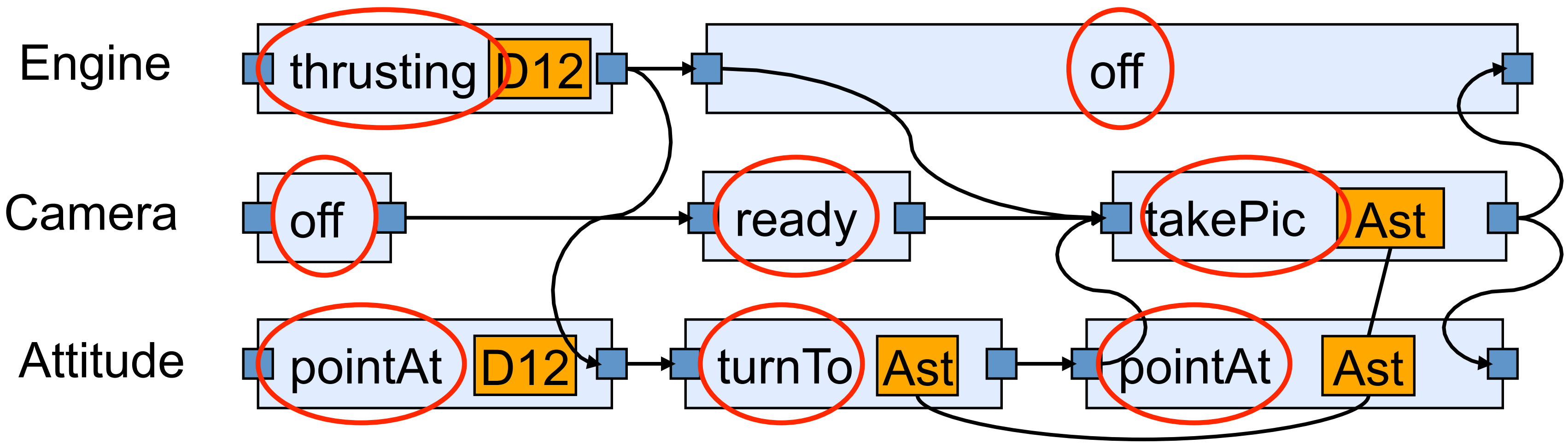
Constraint-based plan

- Plan is a network of intervals representing activities



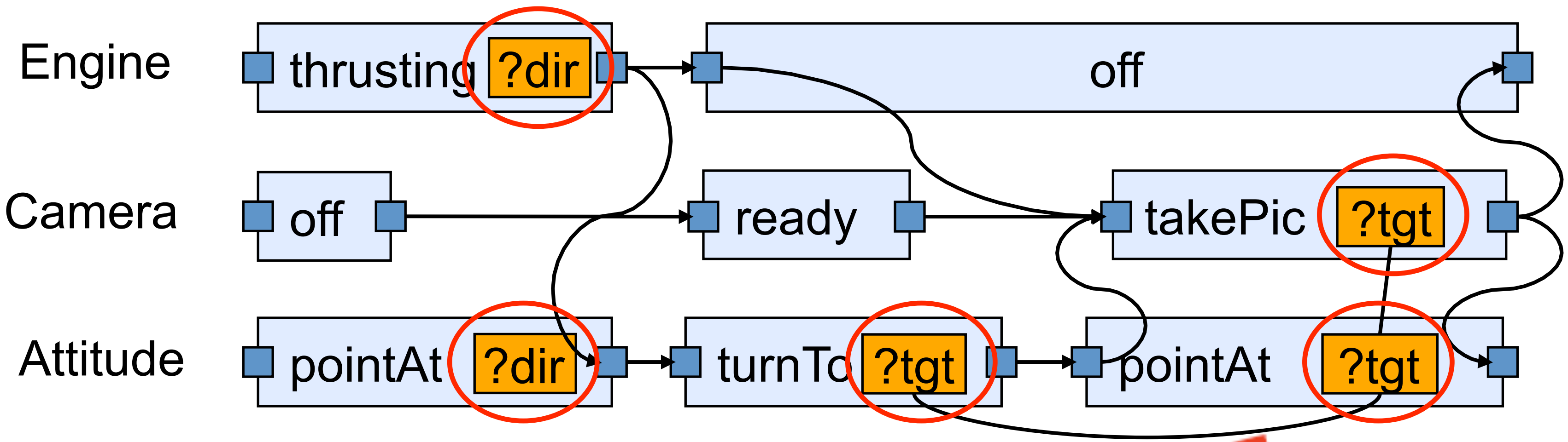
Predicates

- Logical predicates describe actions and states



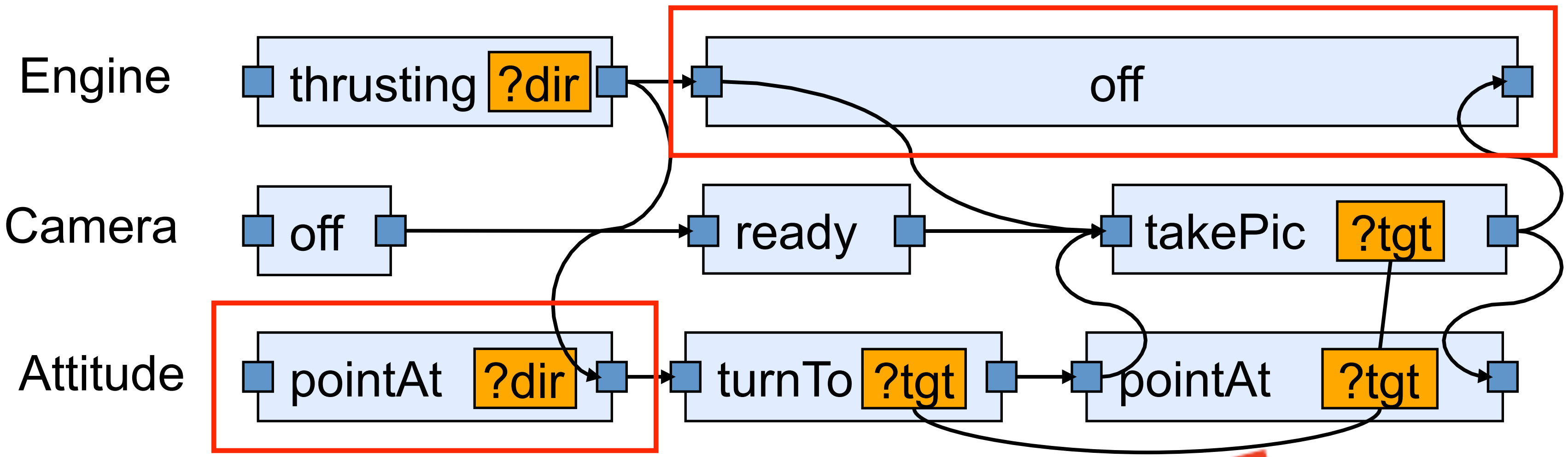
Predicate Parameters

- Parameterized predicates
  - Each predicate type has a fixed set of parameters
  - Each parameter instance comes from associated domain
  - Parameters described by variables



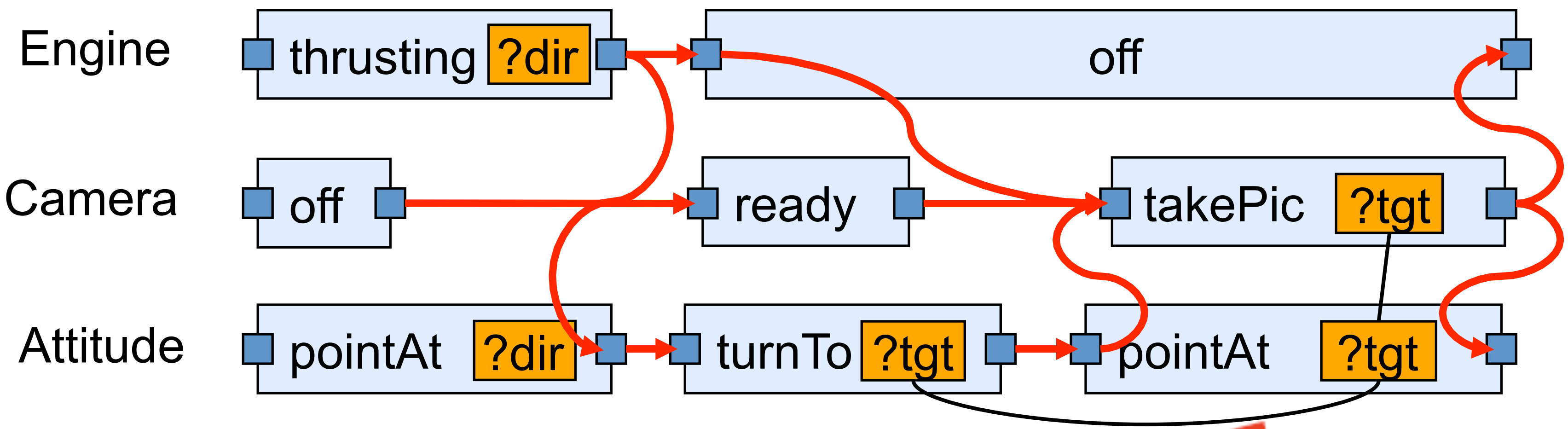
Intervals

- Interval describes activity with duration
  - Start and end times (   )
  - Predicate and parameter variables



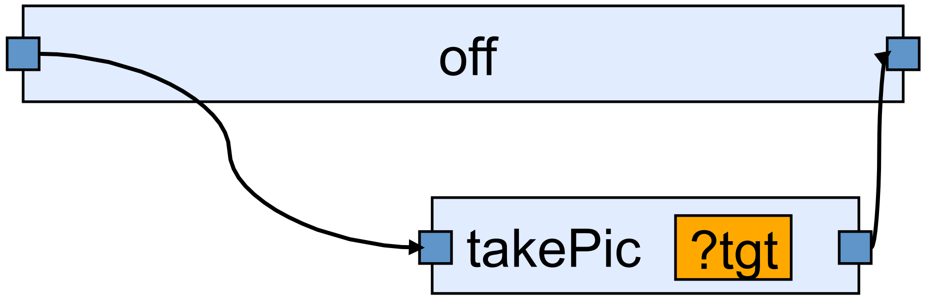
Temporal constraints

- Temporal relations among intervals
  - Can be represented as constraints among start/ end times



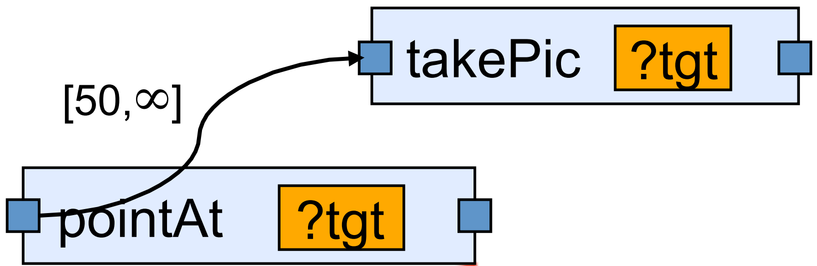
Temporal Constraints

- Qualitative relations
  - before,after, contains,contained by,
  - ....
  - Example: takePic contained by off



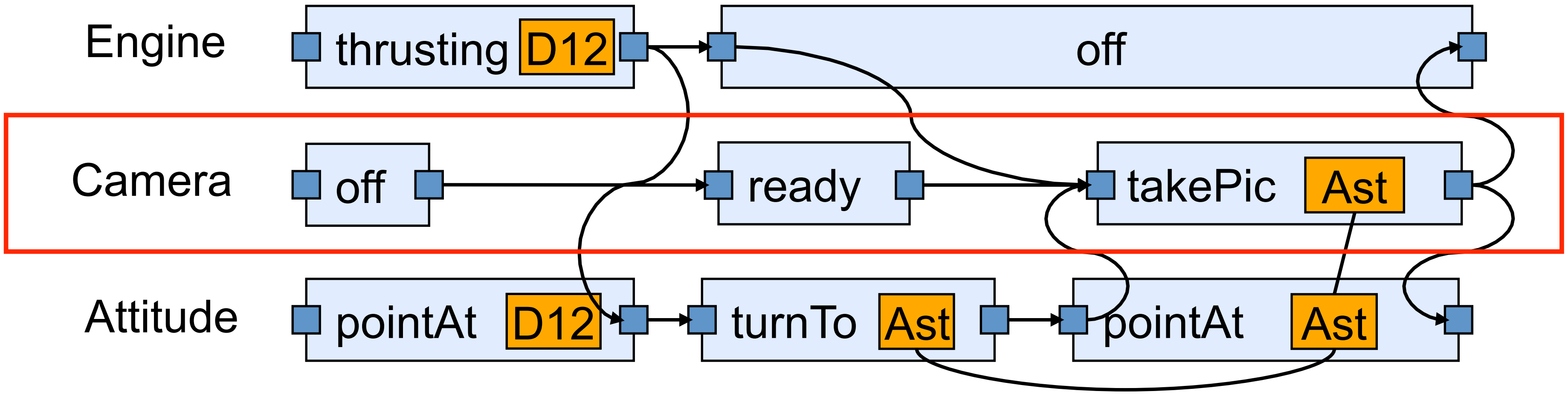
Quantitative bounds

- Example: pointAt starts at least 50 seconds before takePic



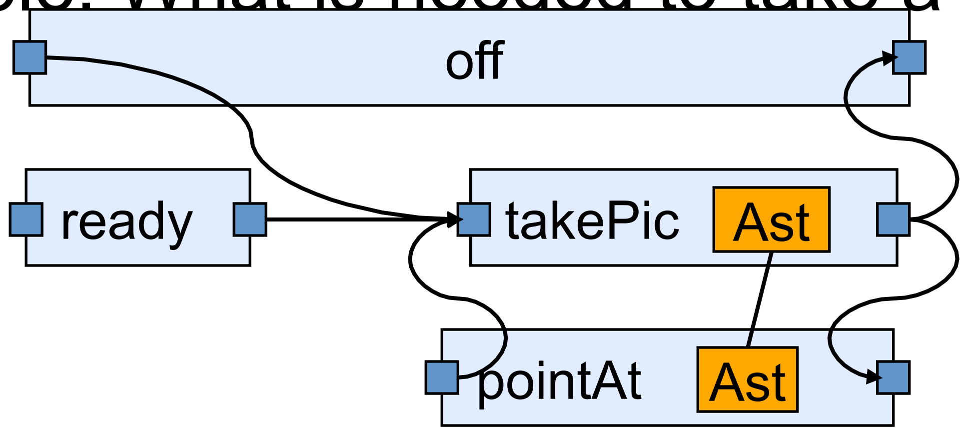
Timelines

- Enforce that activities for same system do not conflict
  - Activities on same timelines are temporally ordered



Subgoaling constraints

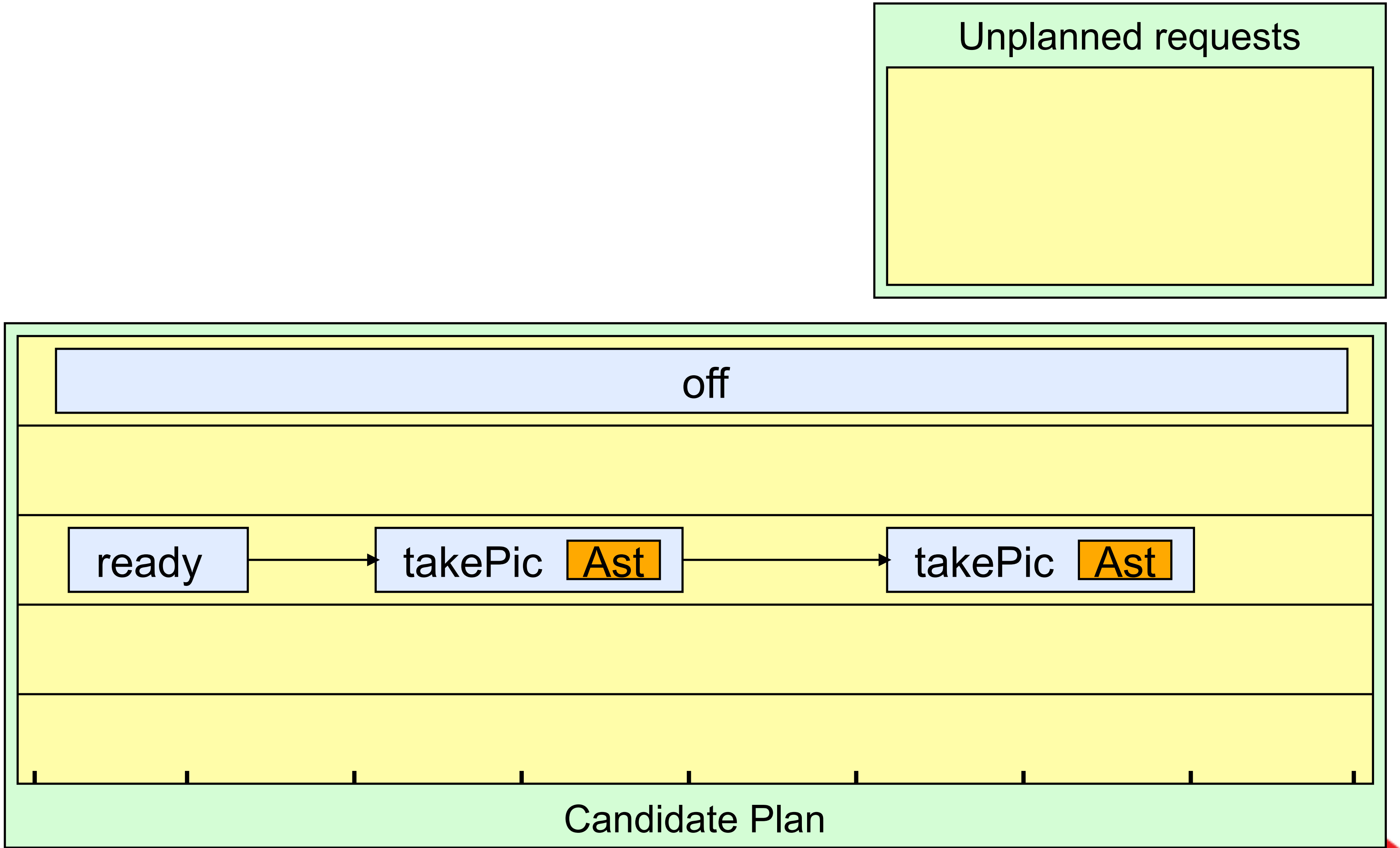
- Subgoals
  - In order to achieve an activity, other activities must happen
  - Example: What is needed to take a picture



Subgoaling

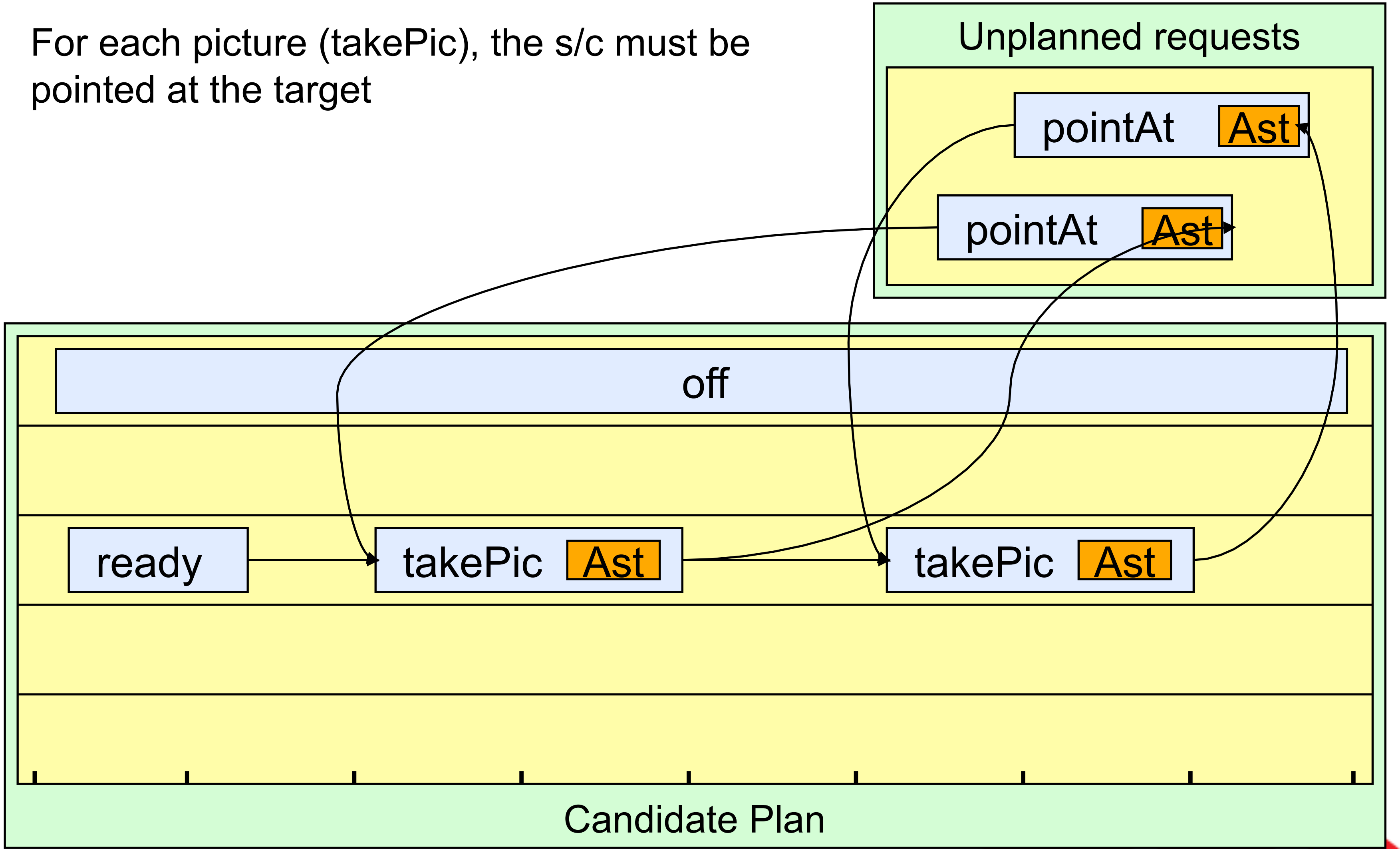
- Automatic addition of necessary subgoals
  - Any activity may give rise to subgoals
    - If activities are merged, only one set of subgoals needed
  - Information from model determines which subgoals are necessary
  - Set of subgoals is affected by parameter variable instantiations
    - Variables affecting subgoals are "decision variables"
- Automatic removel of no-longer-applicable subgoals
  - If the last of a merged set of activities is unscheduled, all associated subgoals are removed from the plan
    - Removal of subgoals then done recursively
  - If a decision variable is assigned values no longer specifying a given subgoal, the subgoal is removed

Simple partial plan



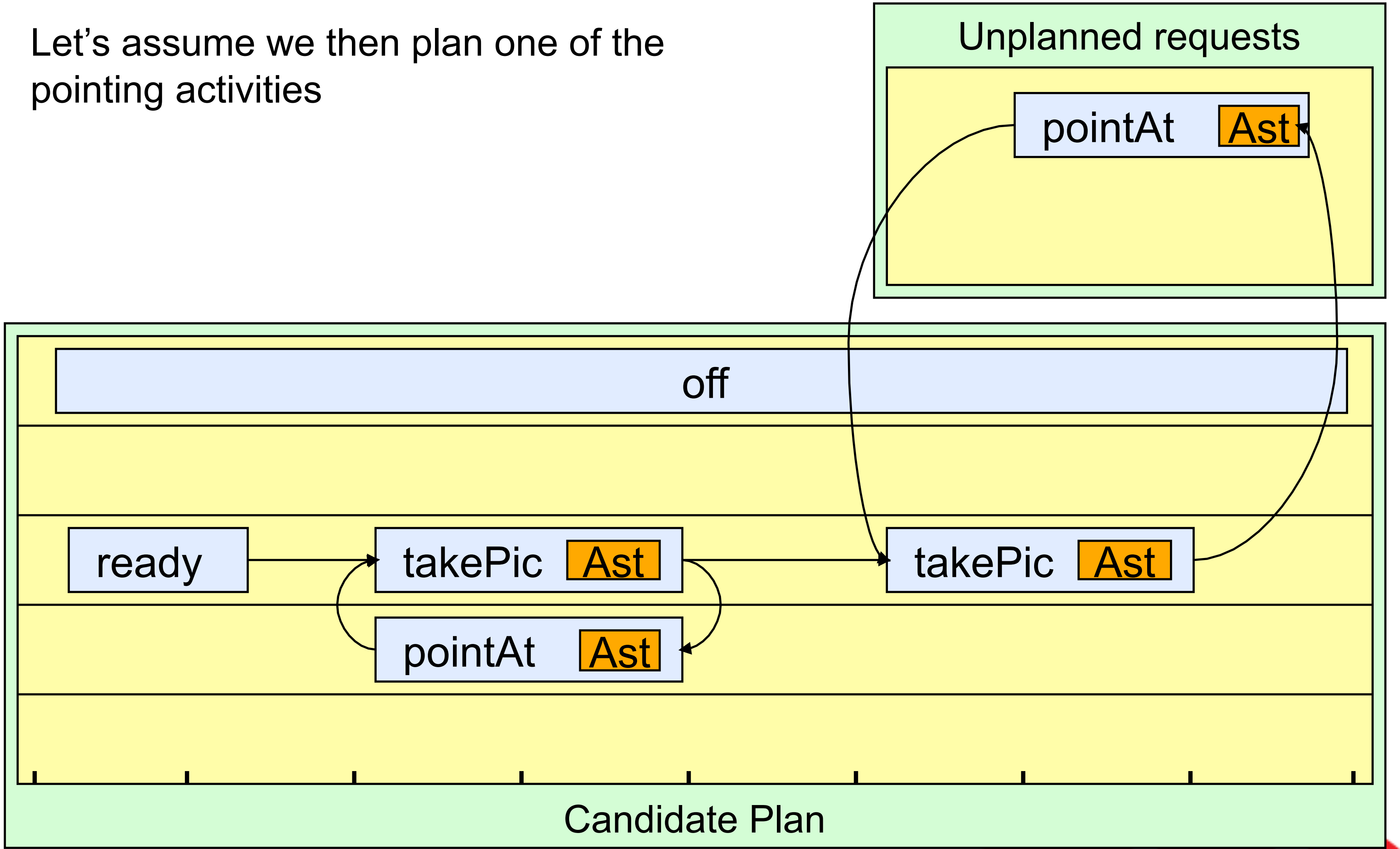
Support activities needed

For each picture (takePic), the s/c must be pointed at the target



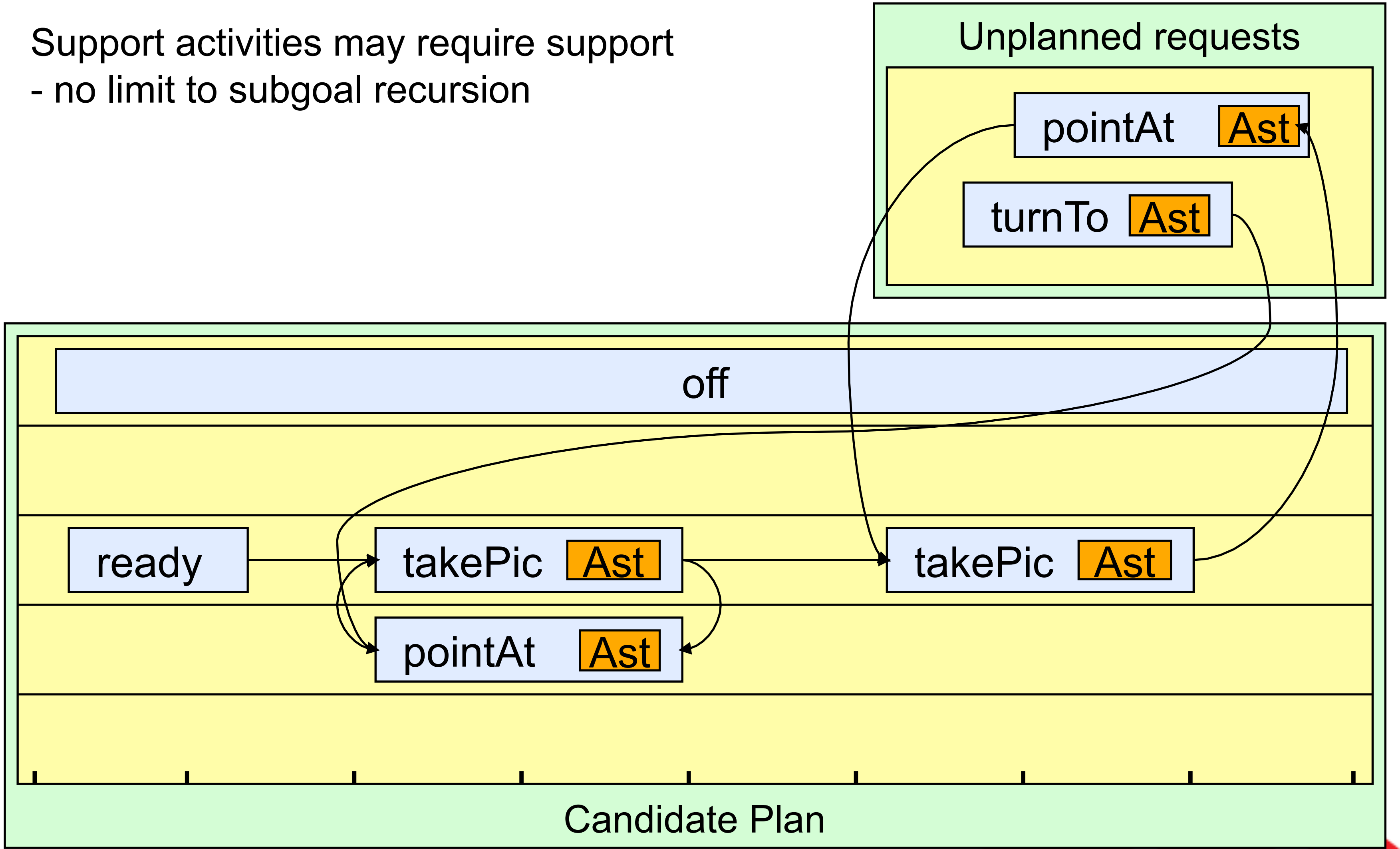
Planning Support Activities

Let's assume we then plan one of the pointing activities



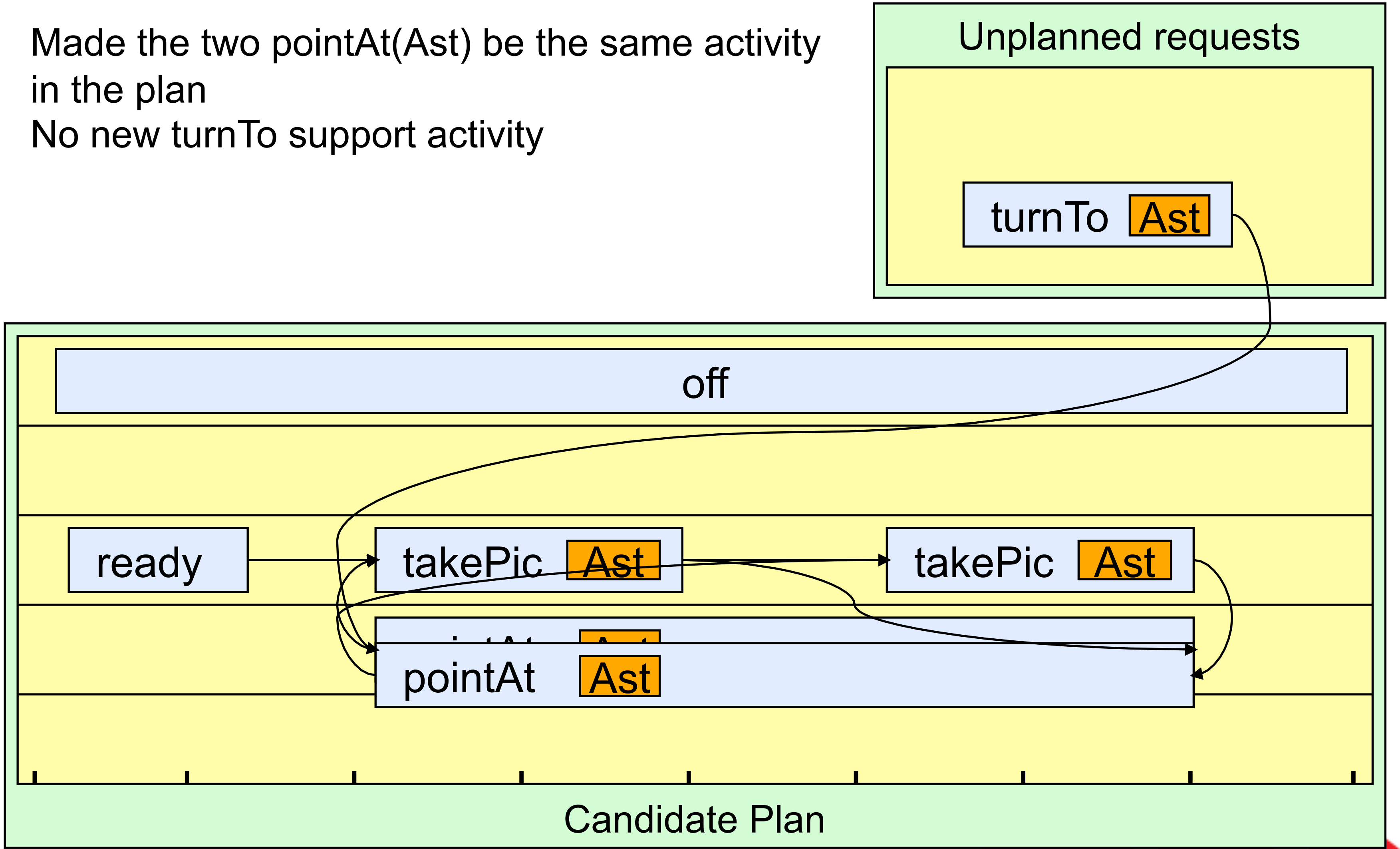
Recursive Support Activities

Support activities may require support
- no limit to subgoal recursion



Only necessary support added

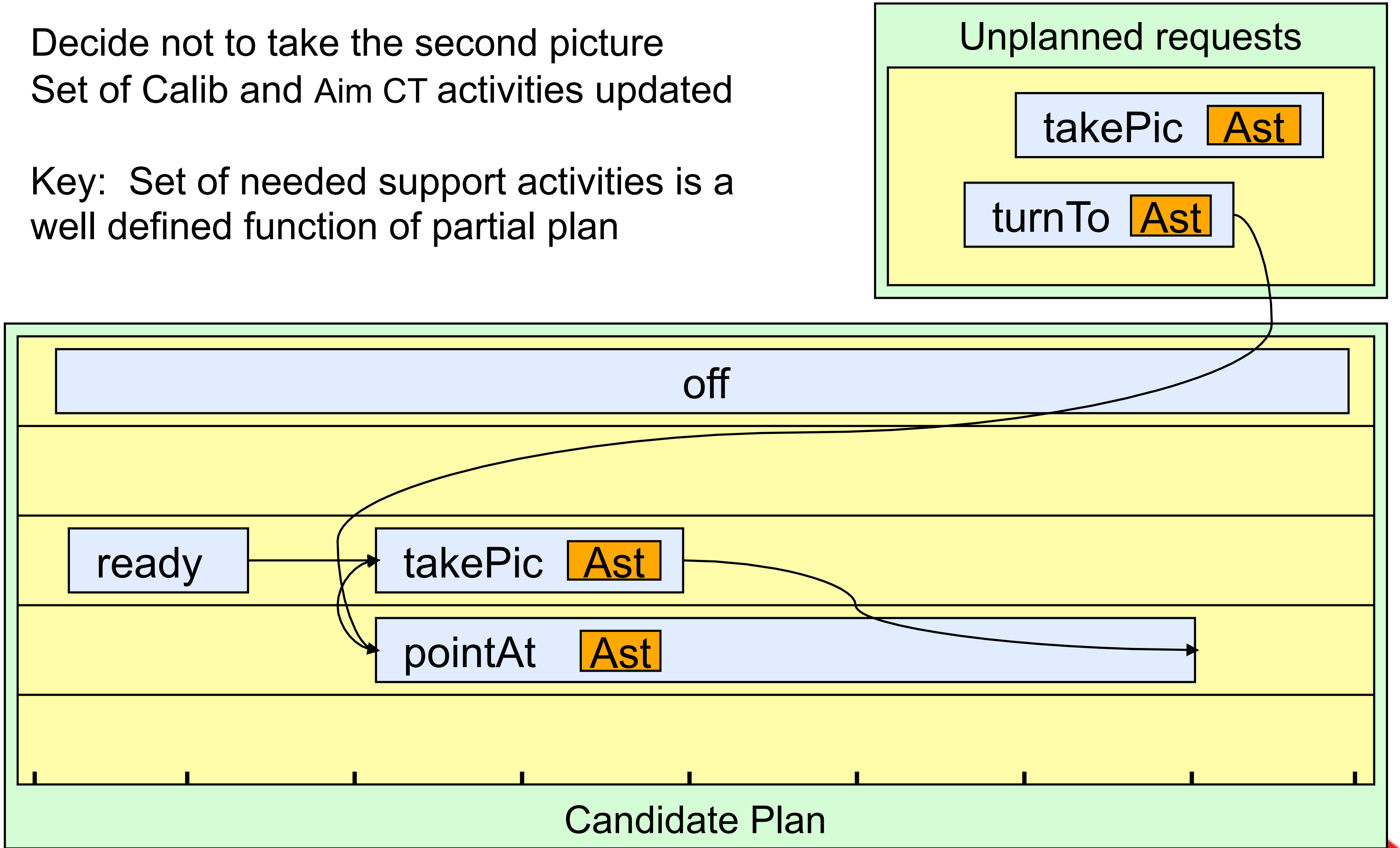
Made the two pointAt(Ast) be the same activity in the plan
No new turnTo support activity



Support requests updated

Decide not to take the second picture
Set of Calib and Aim CT activities updated

Key: Set of needed support activities is a well defined function of partial plan



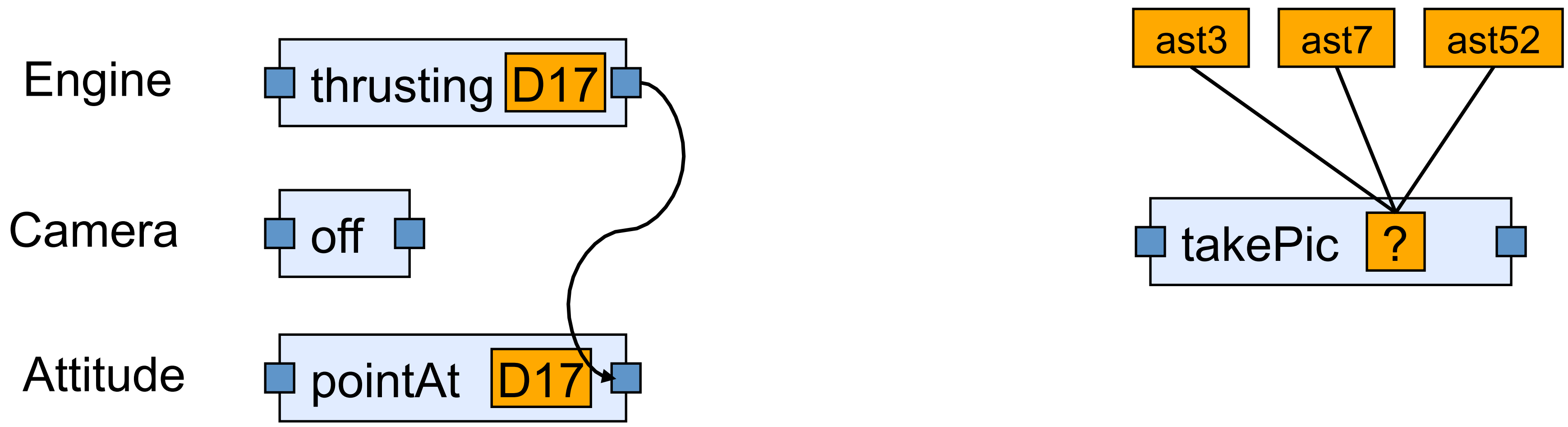
Building Plans

- Automated Search
  - Given: Partial plan, including desired goals
  - Process: Modify candidate plan until valid plan
    - Automated system makes decisions about how to modify plan
  - Result: A complete valid plan, or report one could not be found
- Interactive Plan Construction
  - Human user uses automated reasoning and planning capabilities to collaboratively build a plan
  - Input/output of process the same
  - Process: User and automated system modify plan
    - User makes decisions - automated system handles ramifications
    - User requests help with decisions - automated system put to work
    - Automated system decision overridden by user

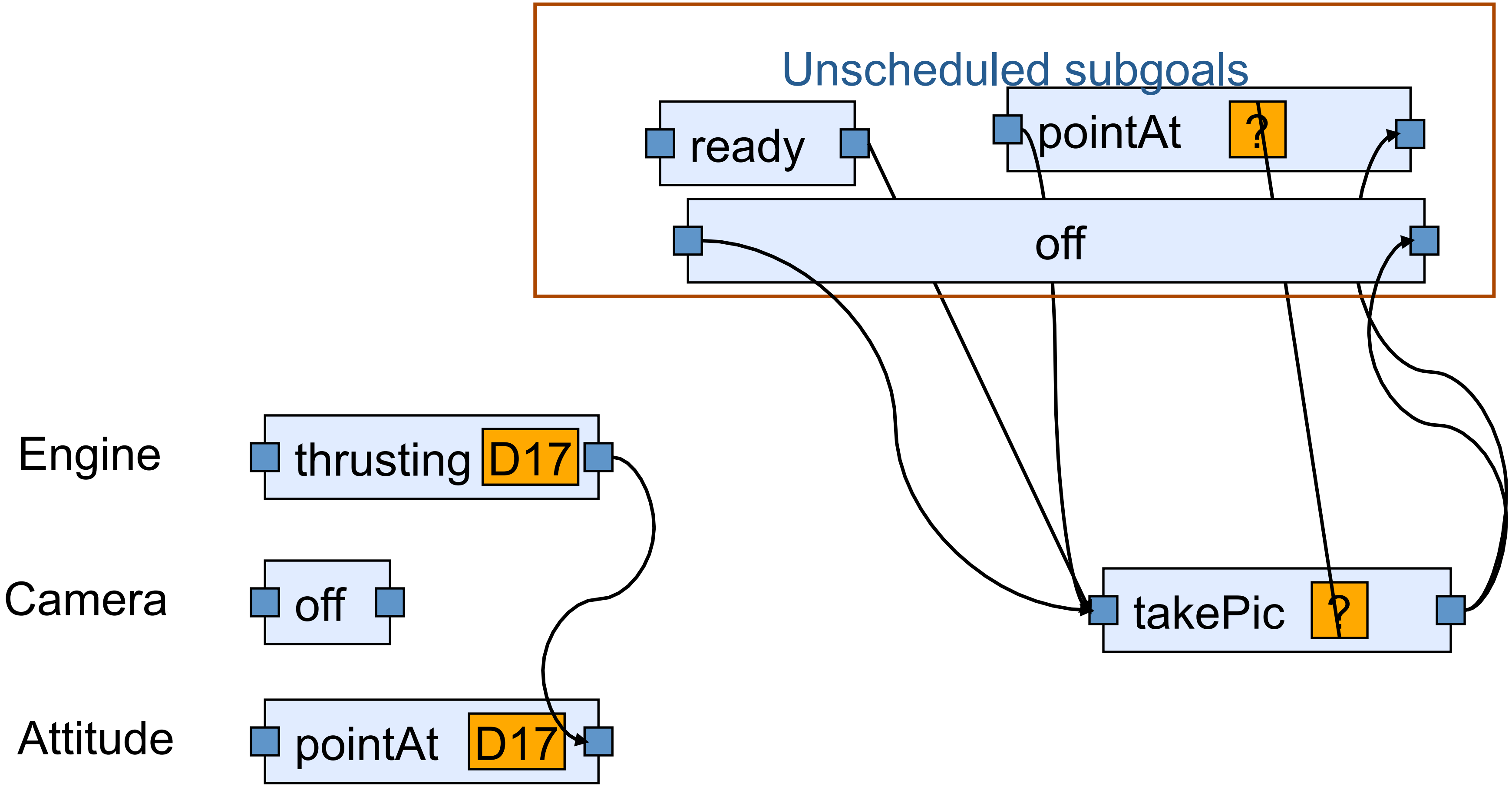
Automated search - outline

- buildPlan(P)
- determine consequences of decisions in P
- if P cannot lead to a valid plan, return failure
- if P is a valid executable plan, return success
- select decision x (parameter or subgoal) in P
- make decision x by assigning a value or interval
- return buildPlan(P + x)

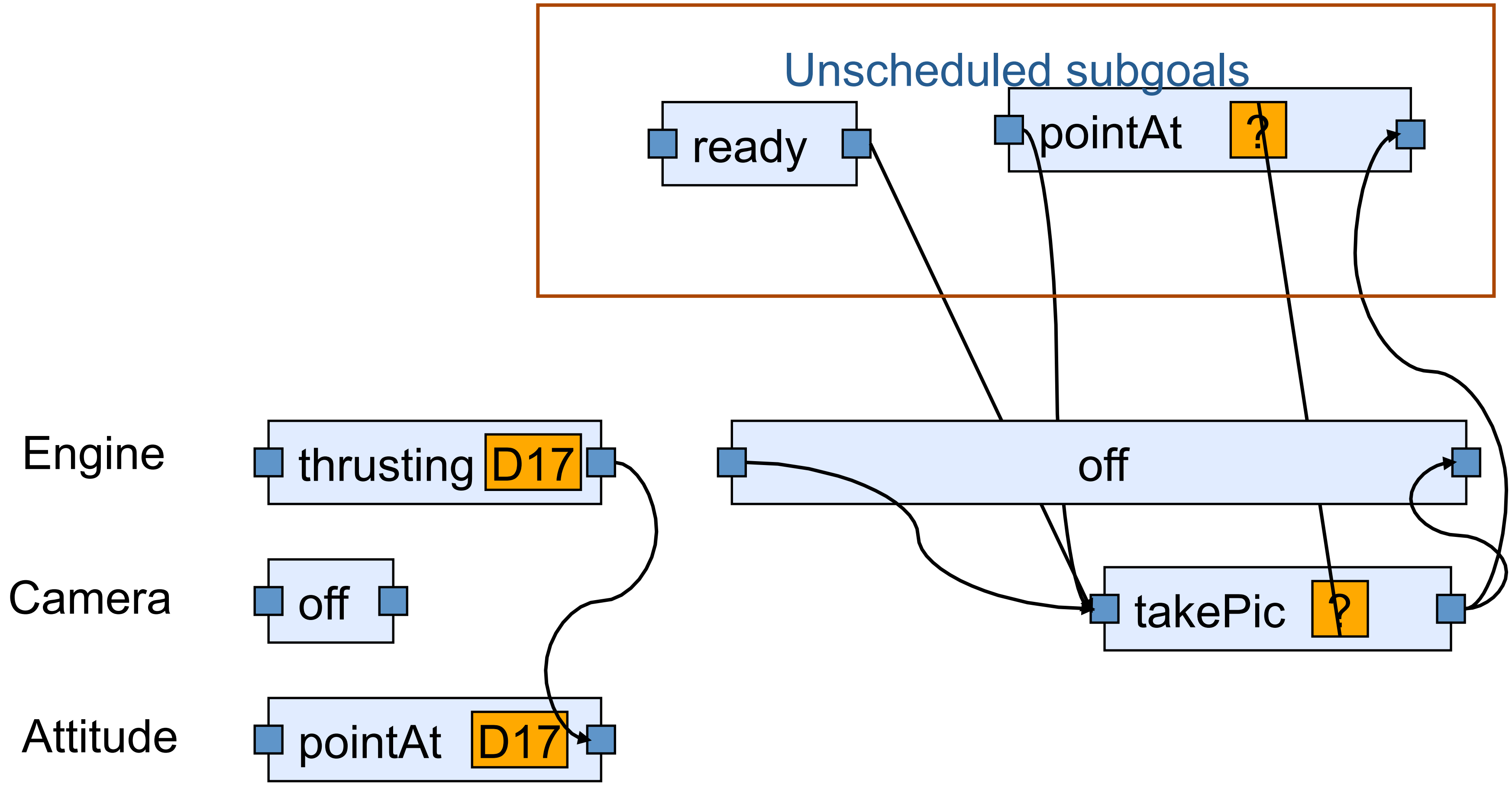
Initial state



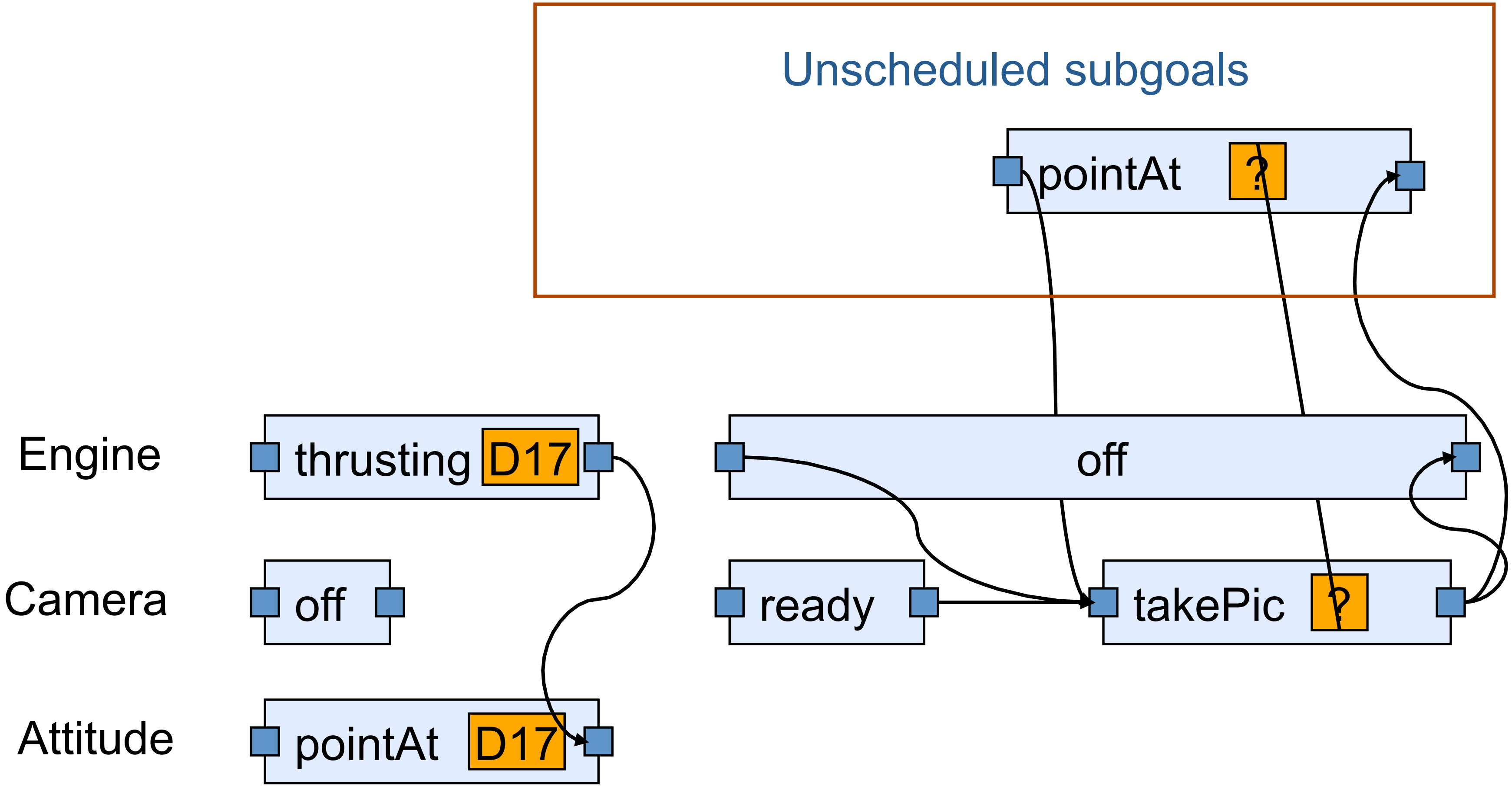
Expand takePic subgoals



Insert *off* subgoal interval

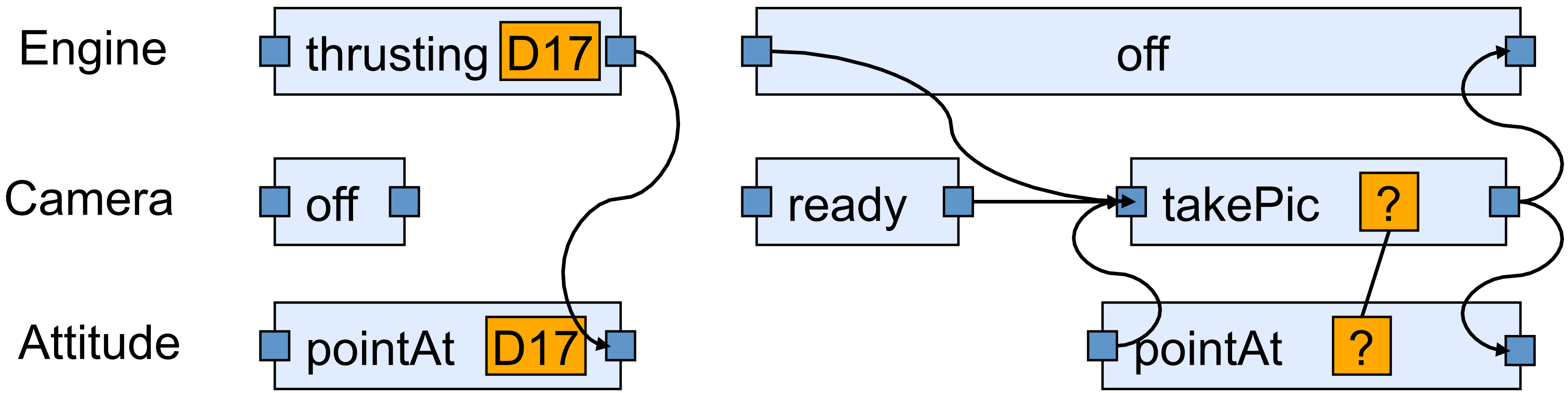


Insert ready interval

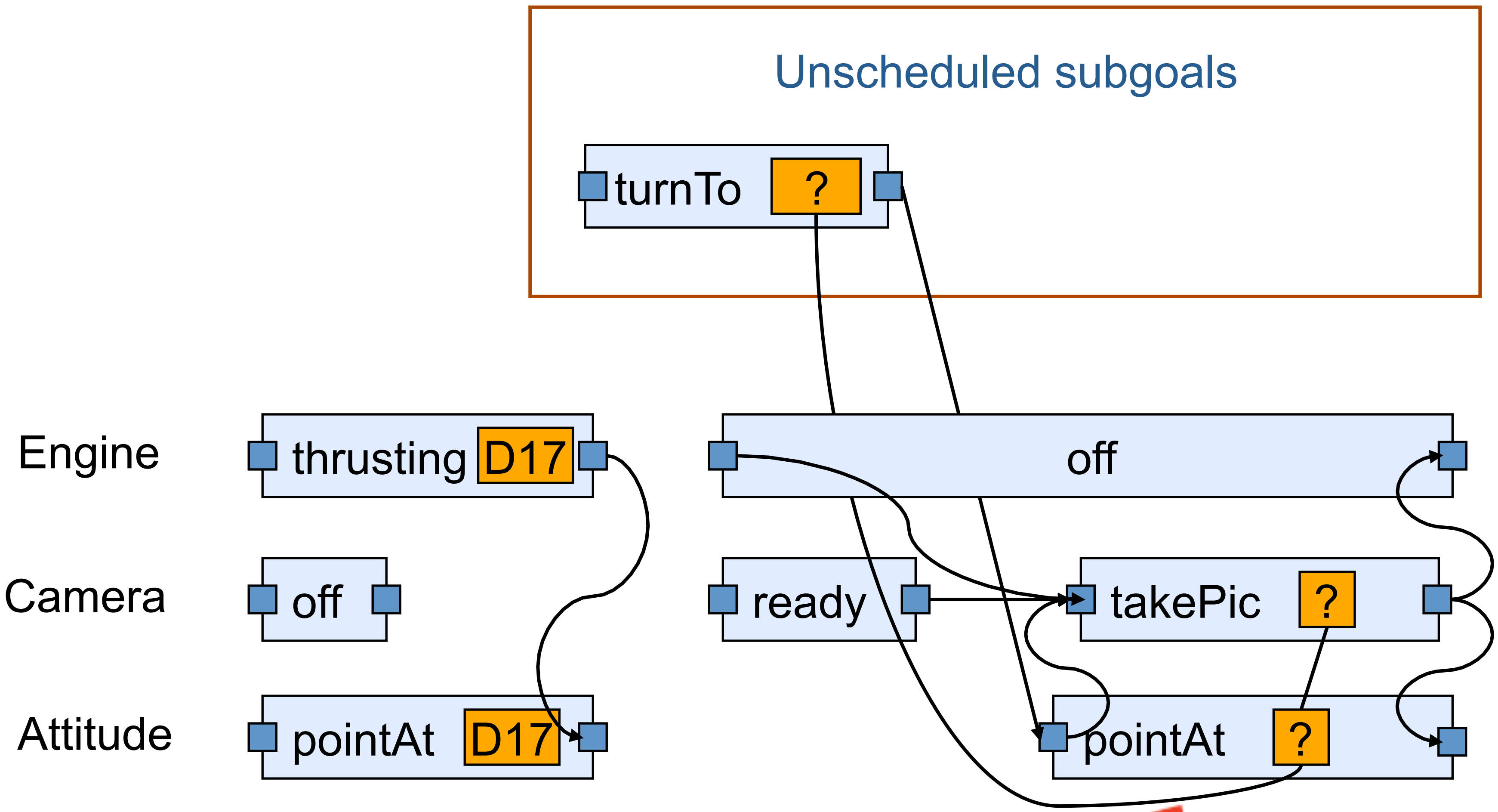


Insert pointAt interval

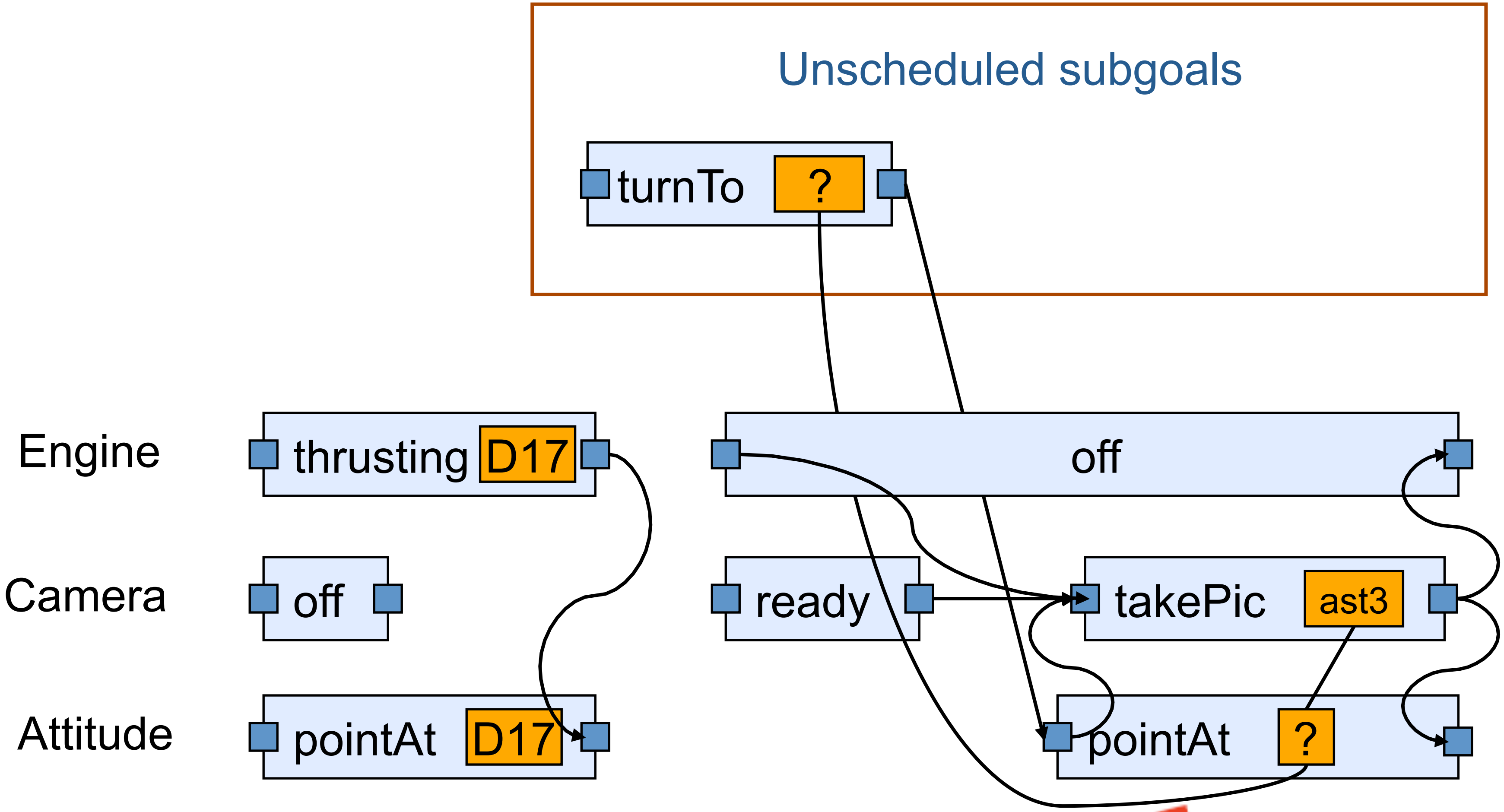
Unscheduled subgoals



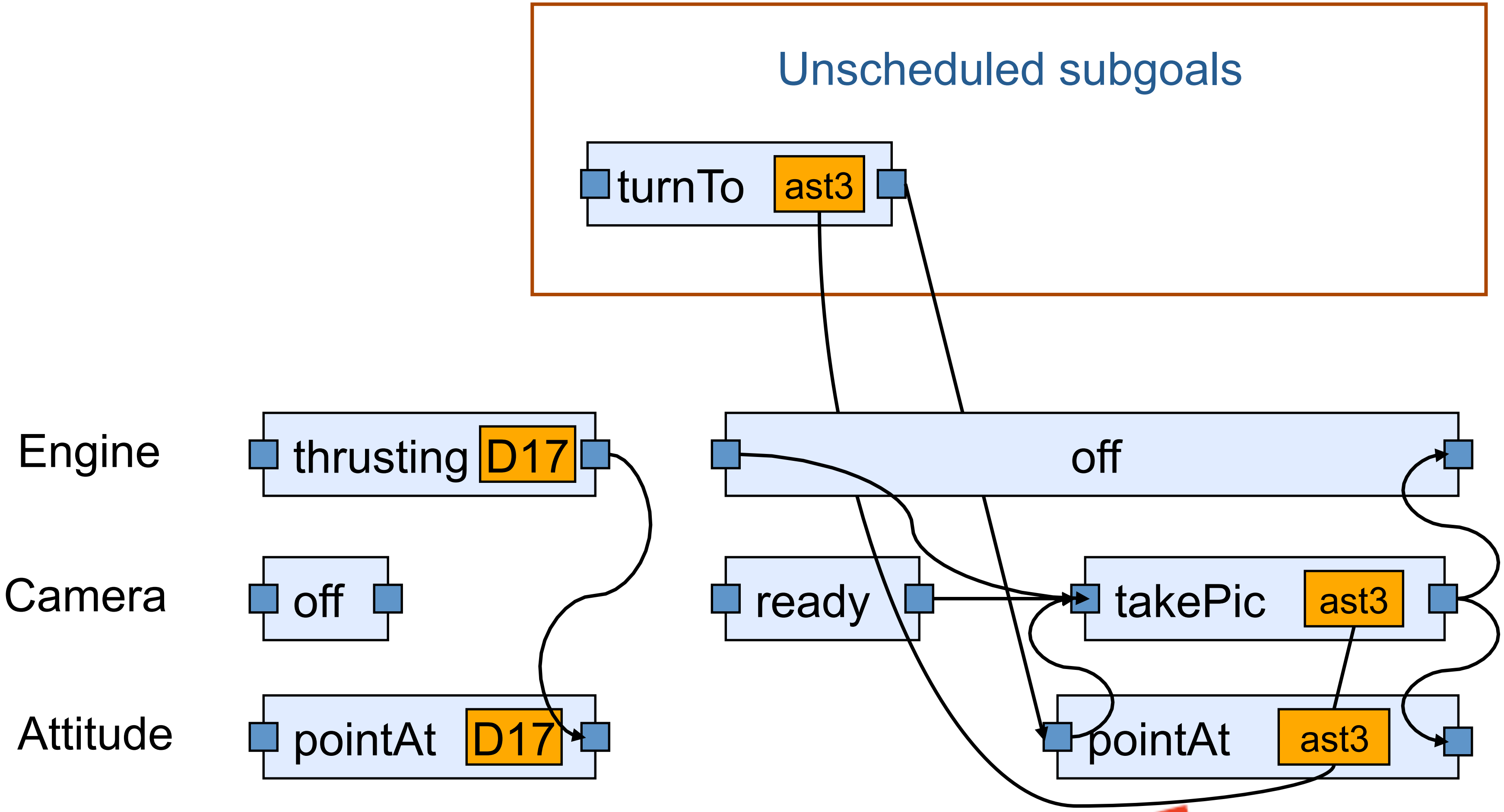
Expand pointAt



Select image target

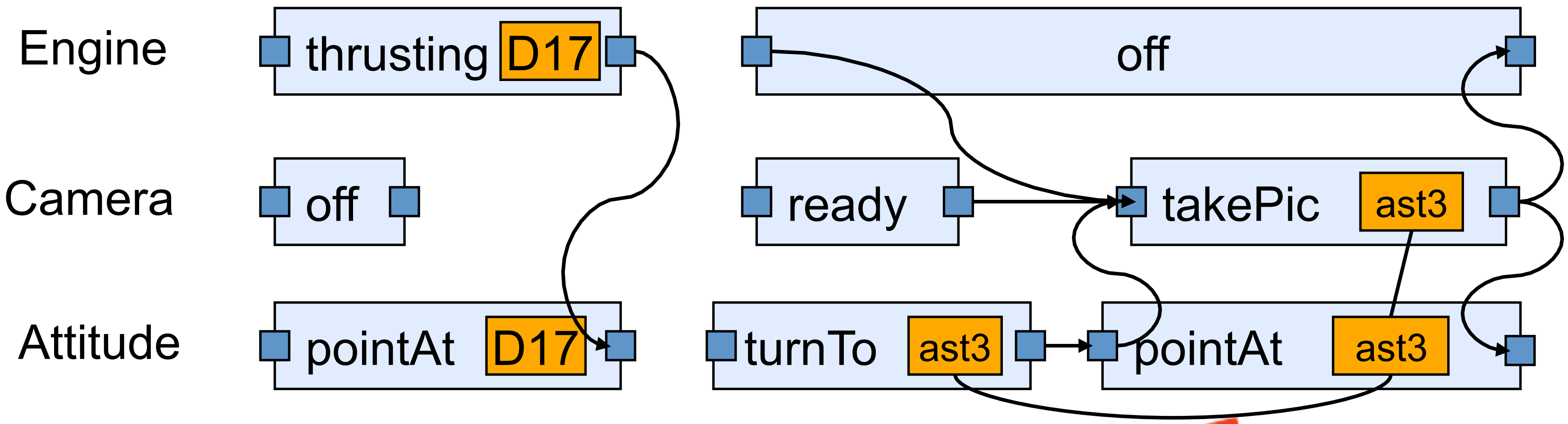


Determine consequences

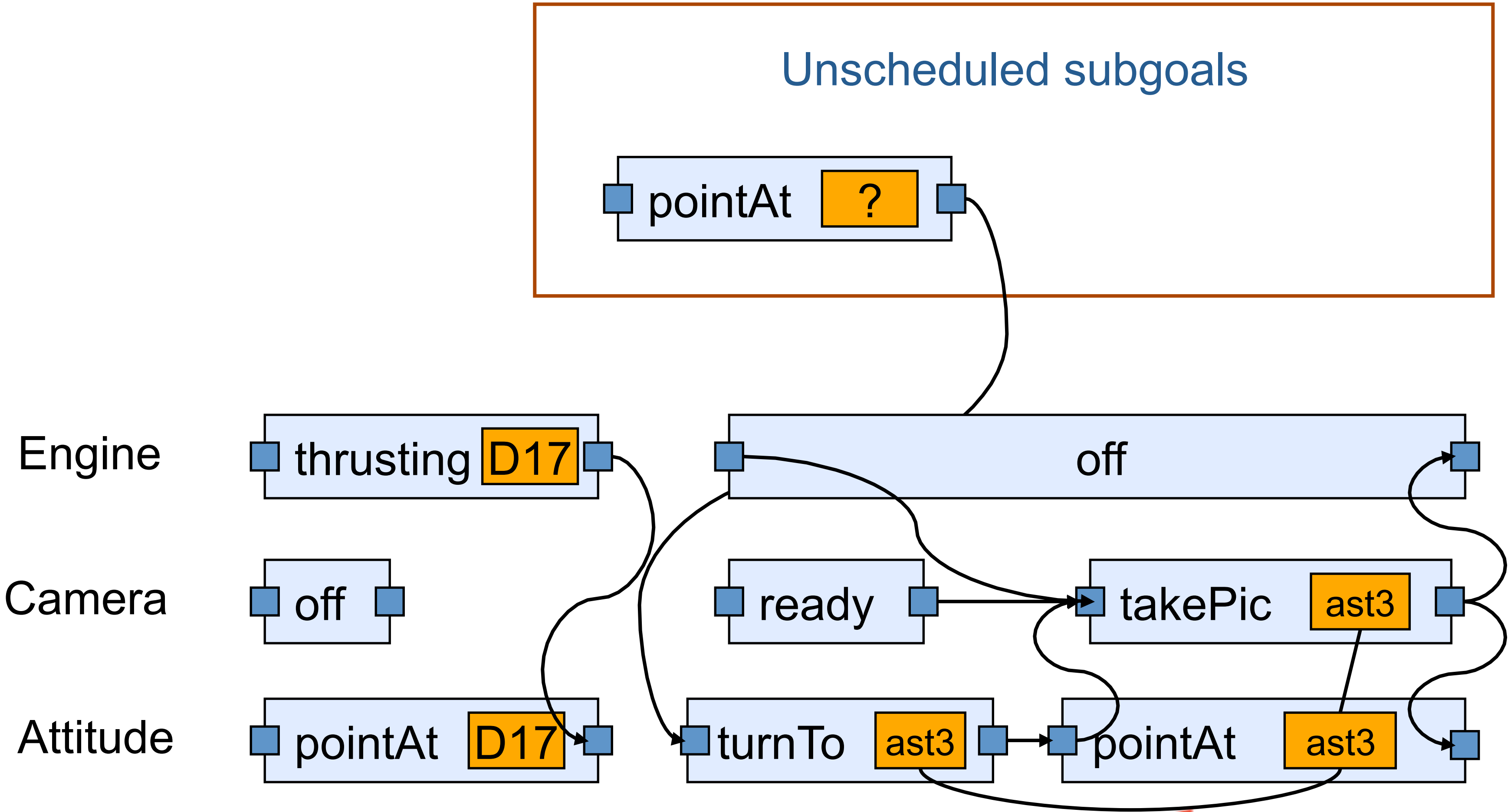


Insert turnTo

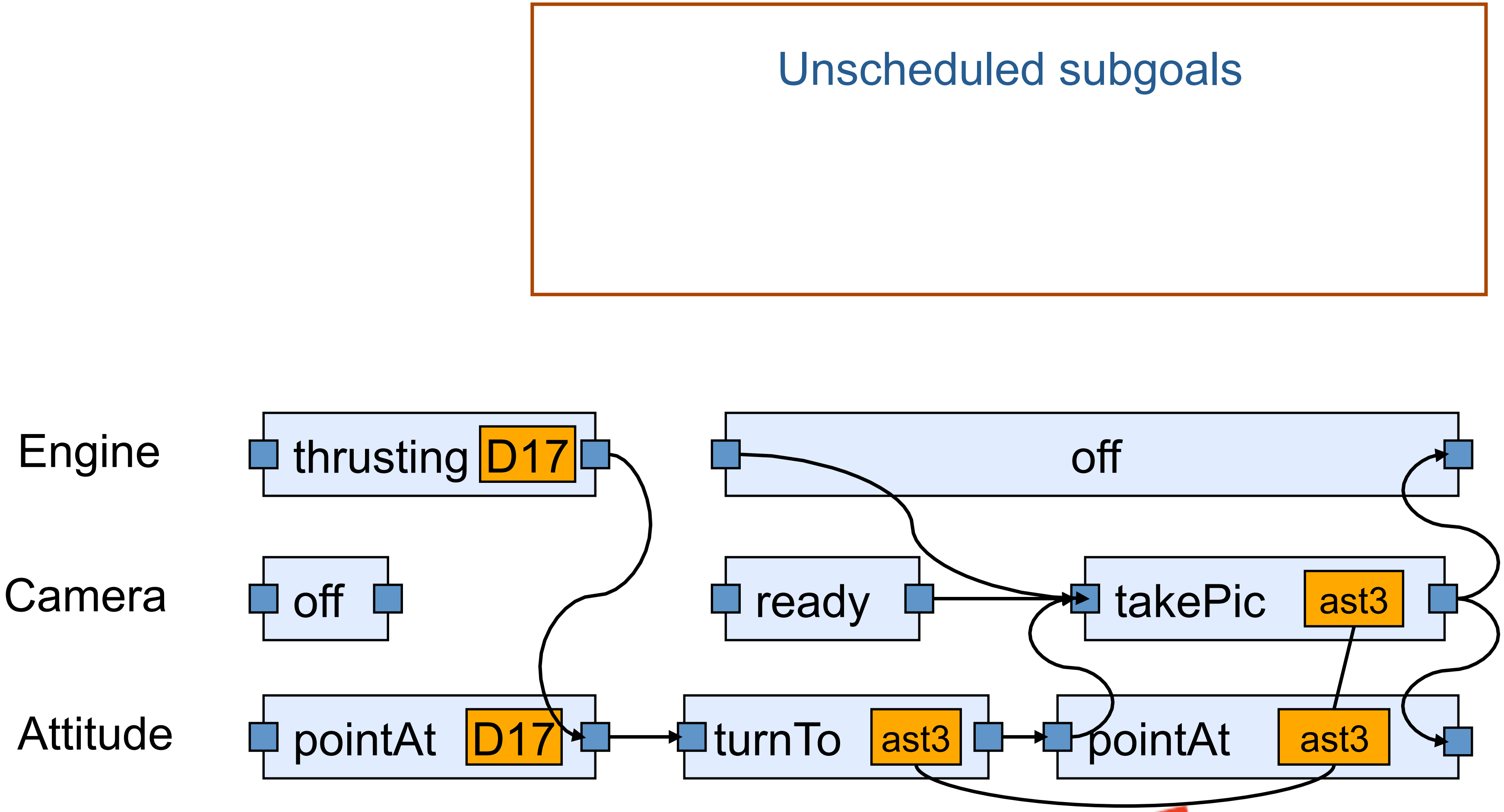
Unscheduled subgoals



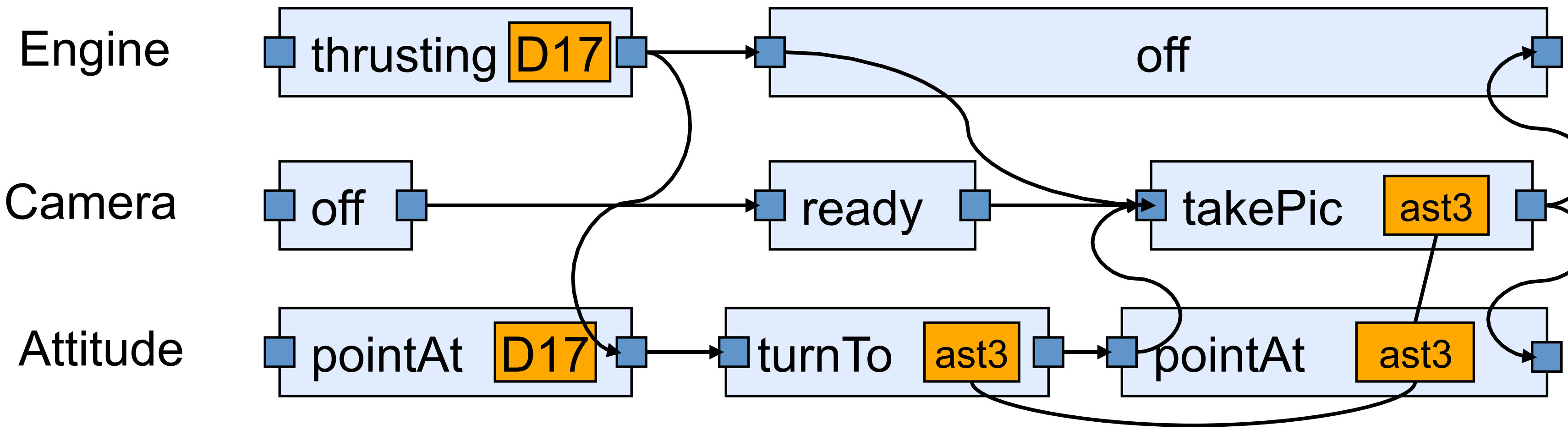
Expand turnTo



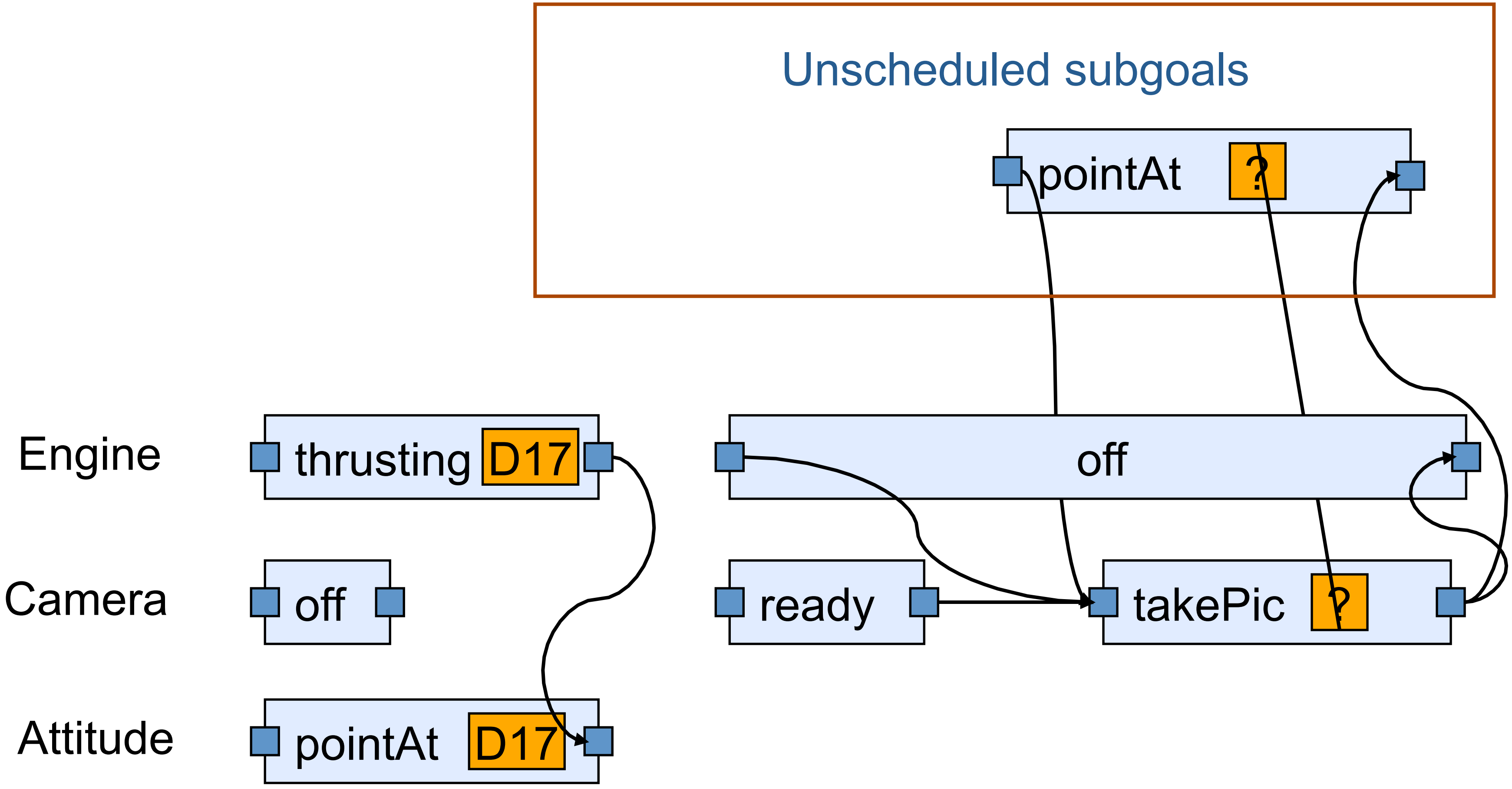
Coalesce pointAt goals



Final plan

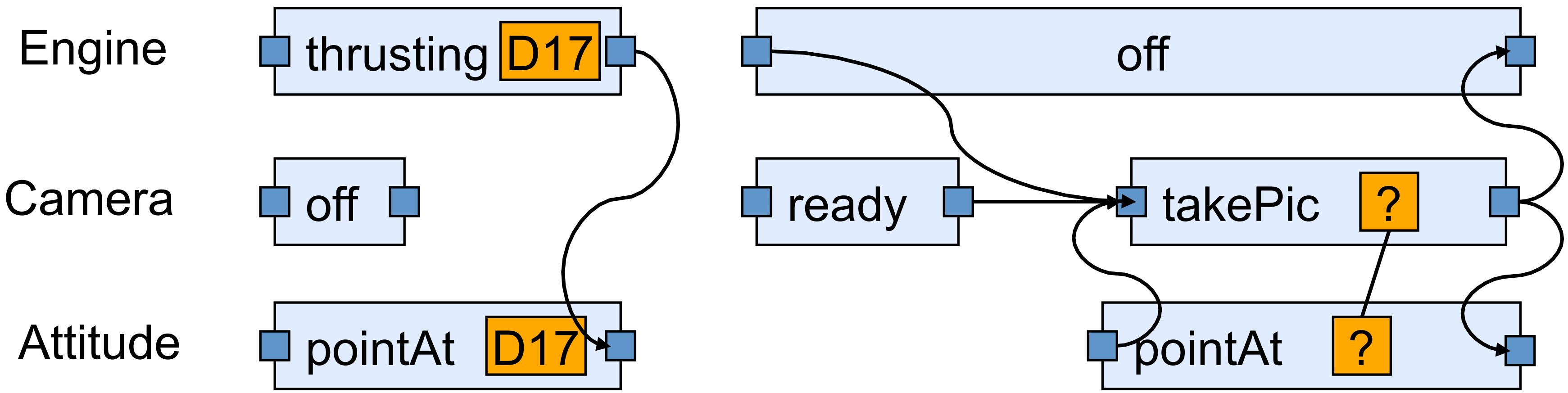

Engine
Camera
Attitude
thrusting
D17
off
off
ready
takePic
ast3
pointAt
D17
turnTo
ast3
pointAt
ast3

Backtracking : can be non-chronological

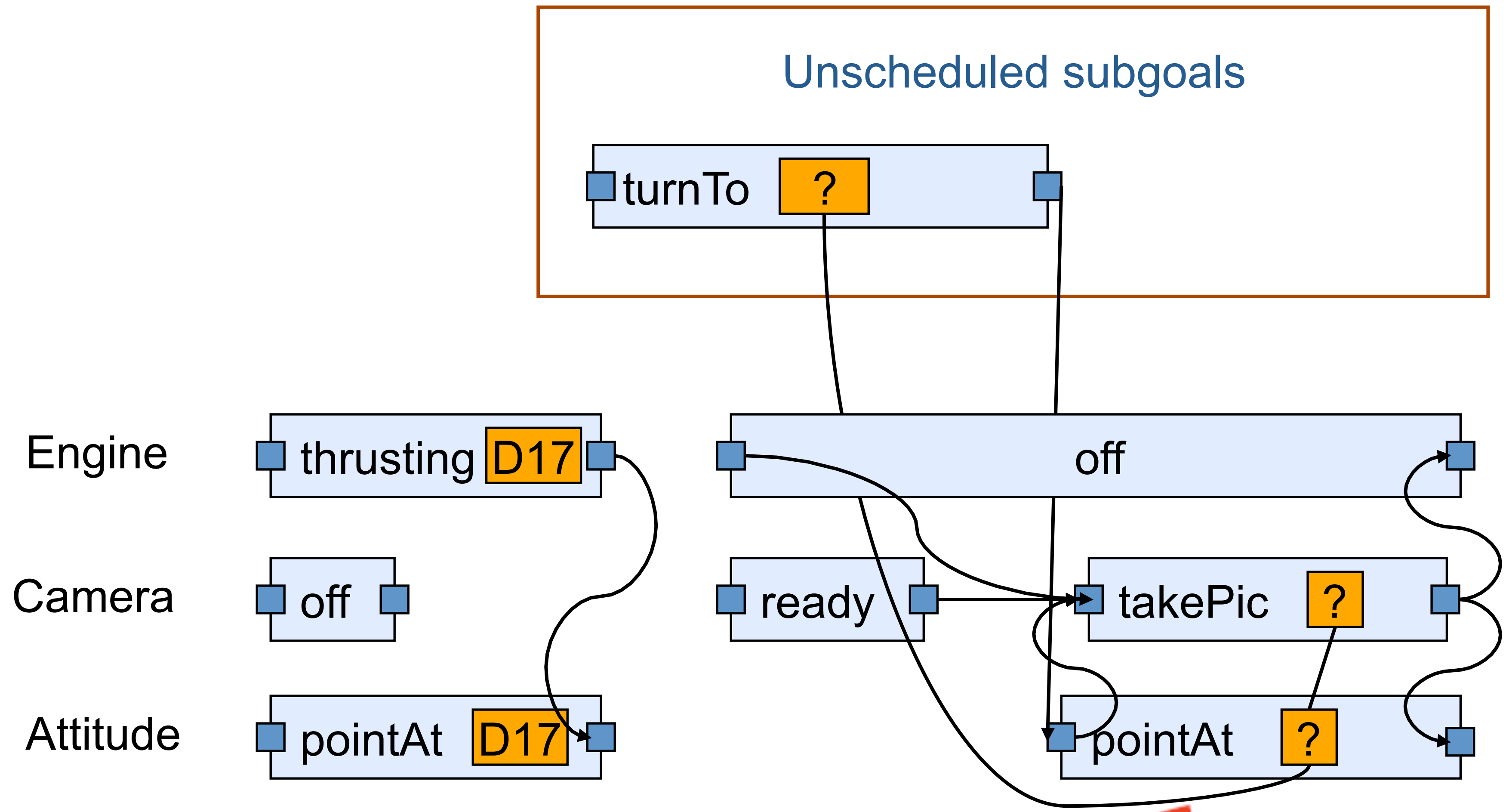


Insert pointAt interval

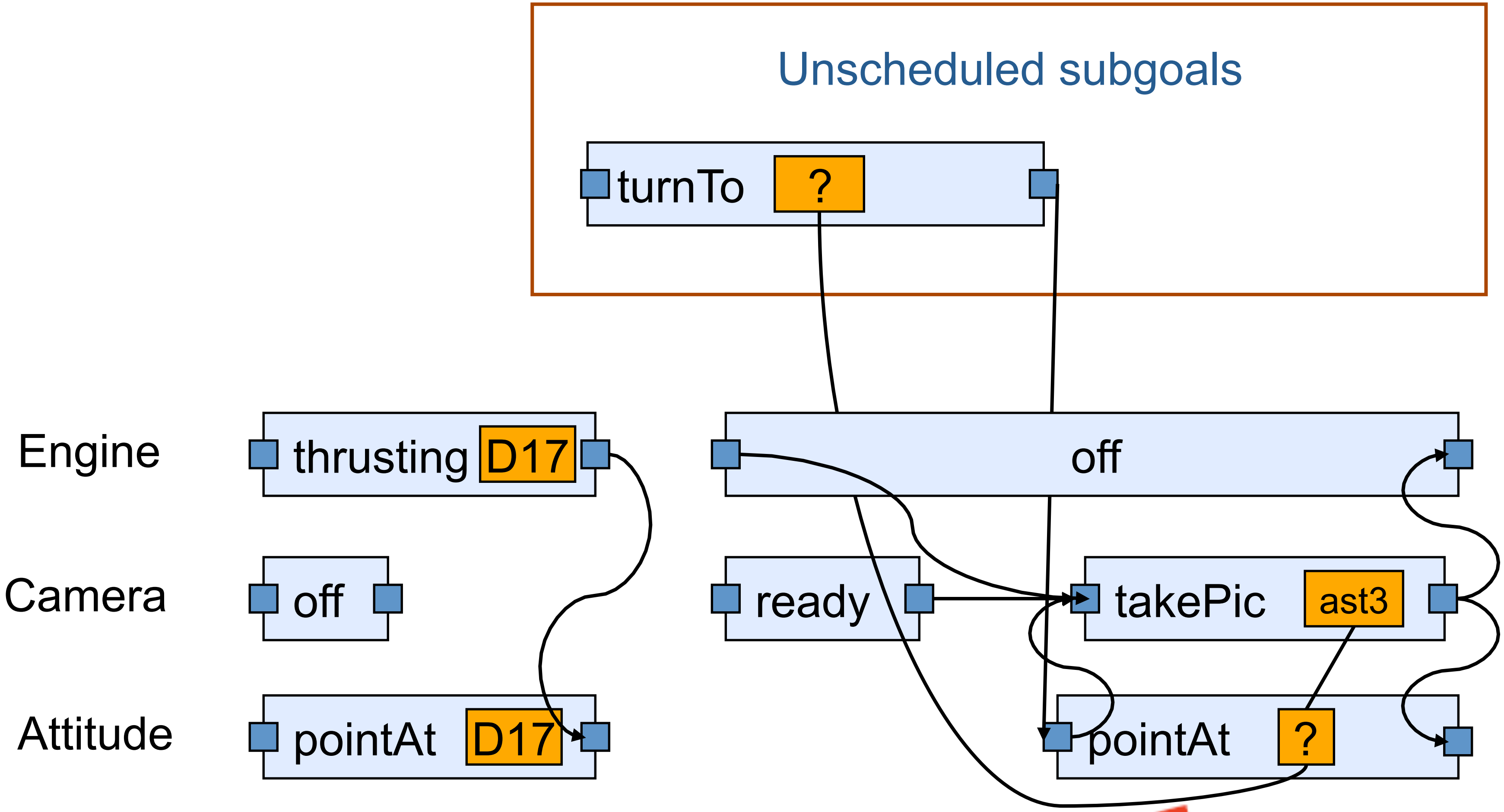
Unscheduled subgoals



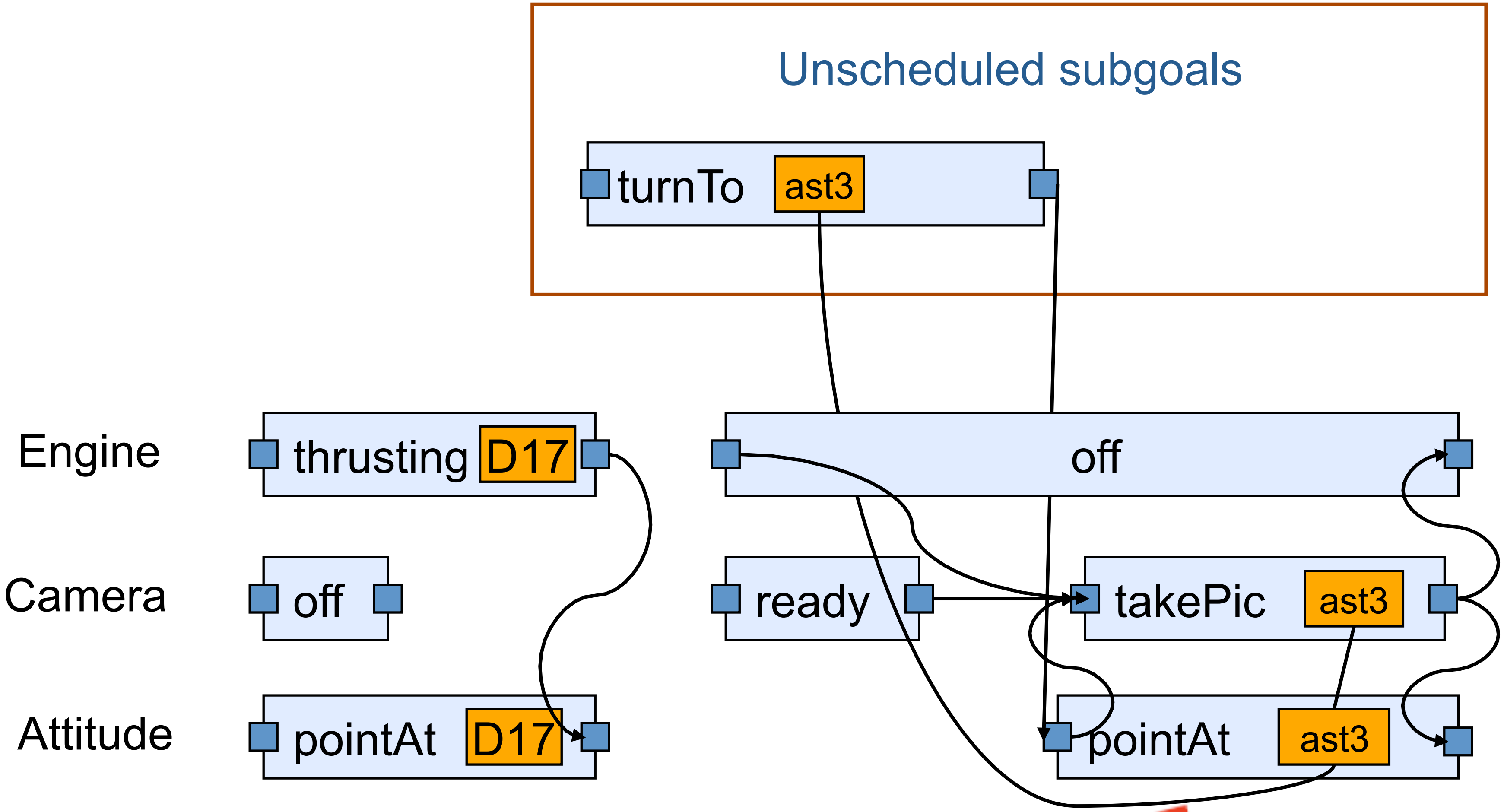
Expand pointAt



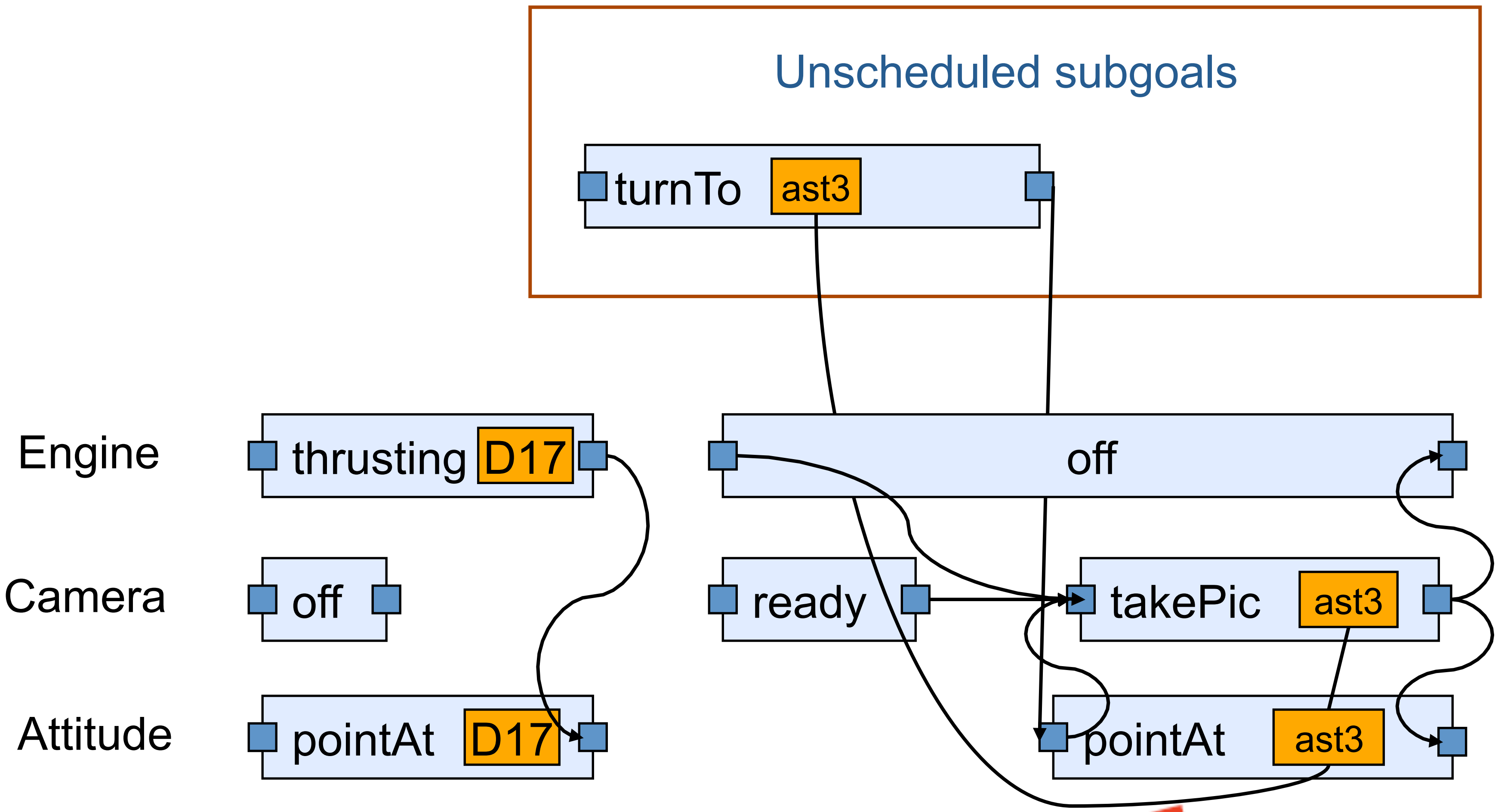
Select image target



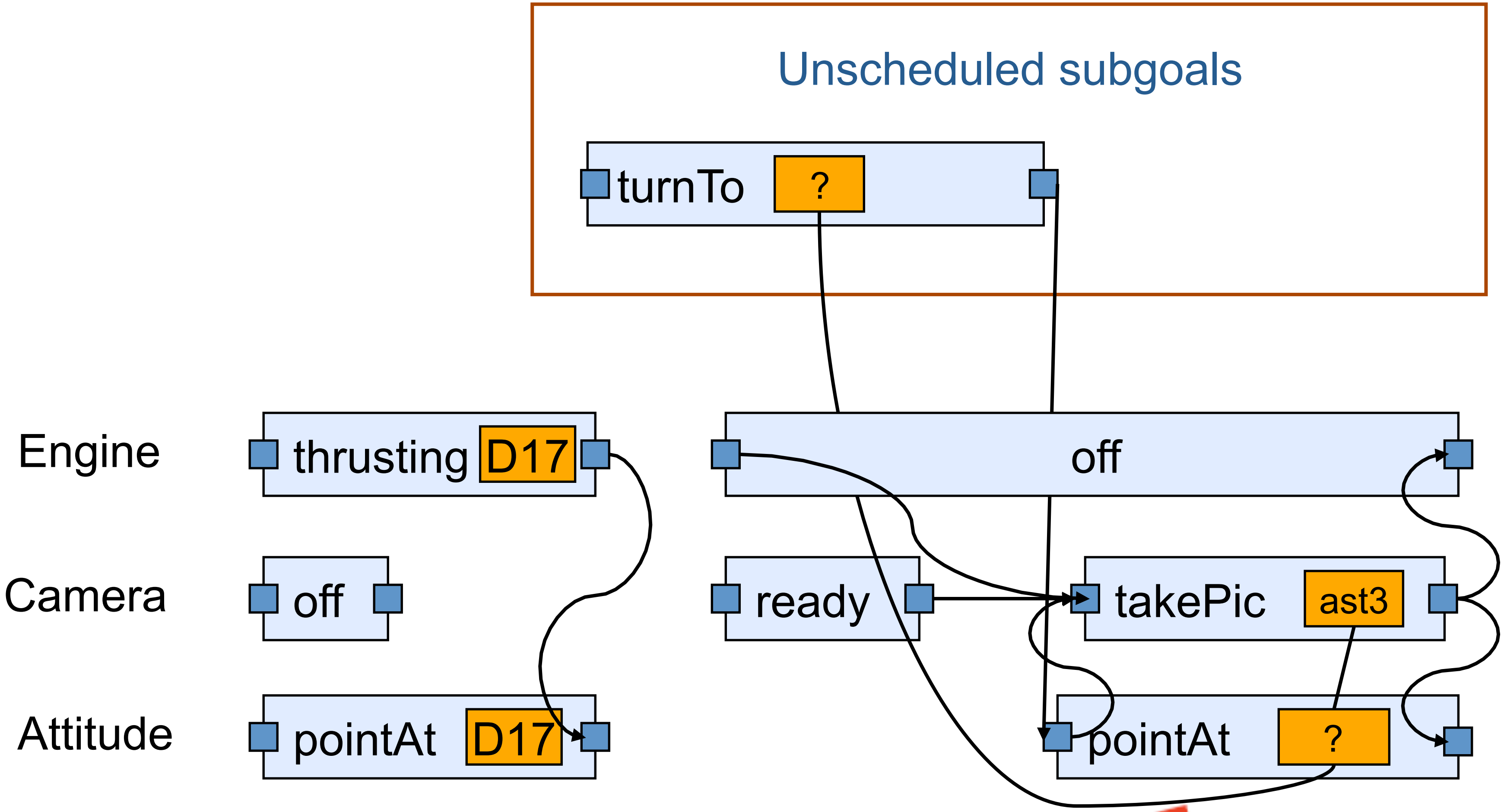
Determine consequences



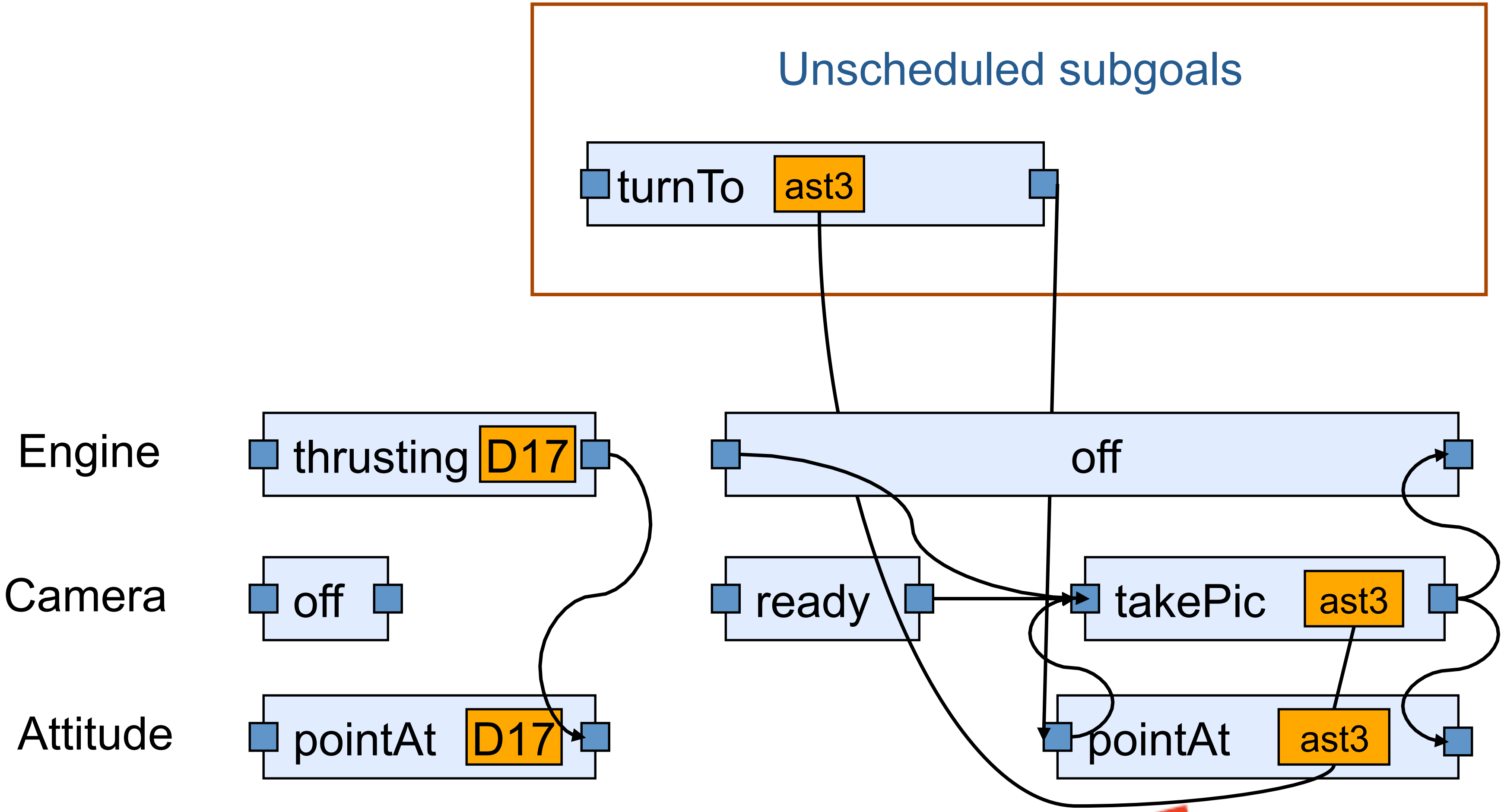
No room for turnTo



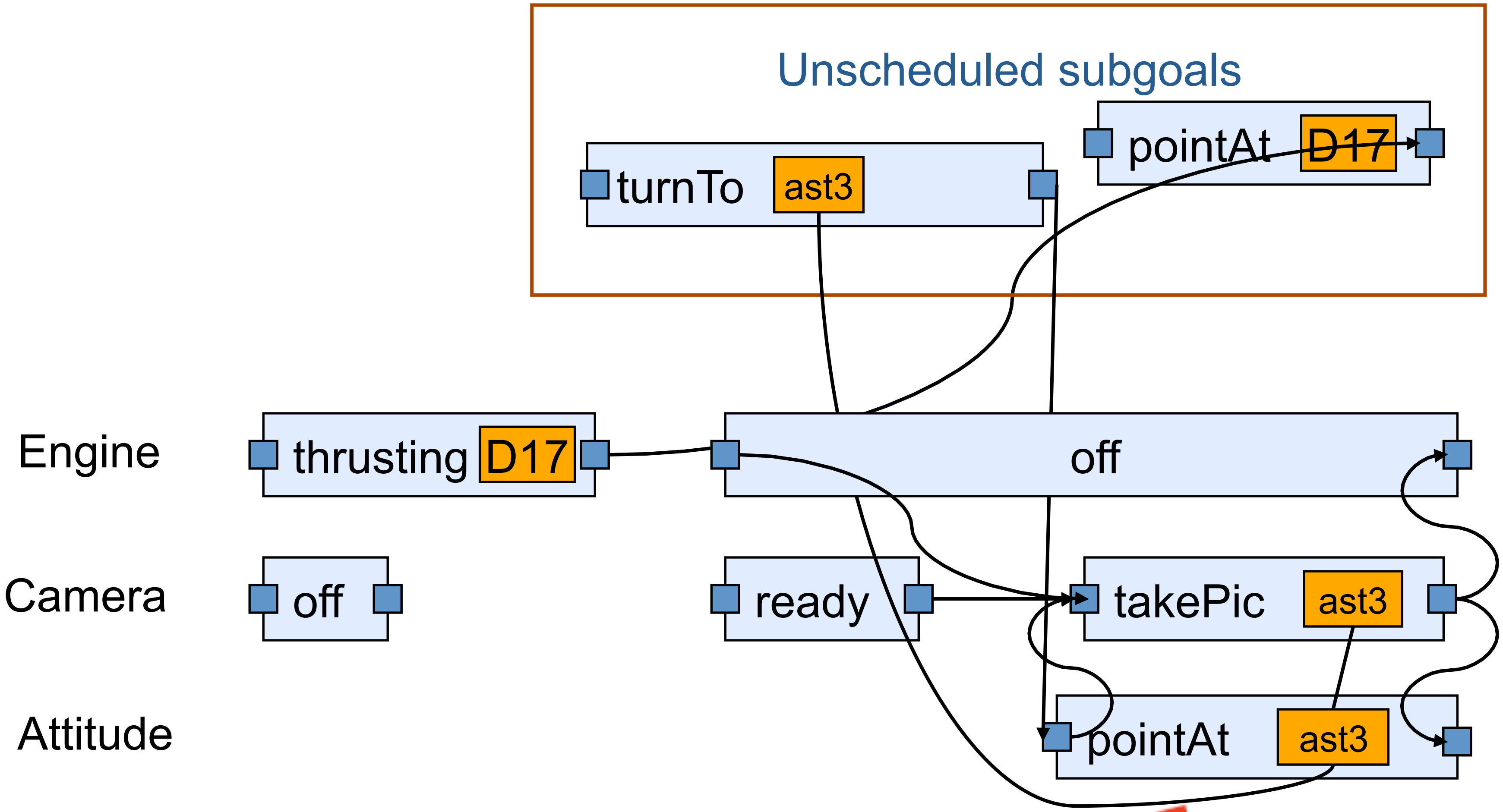
Standard Backtracking - undo last decision



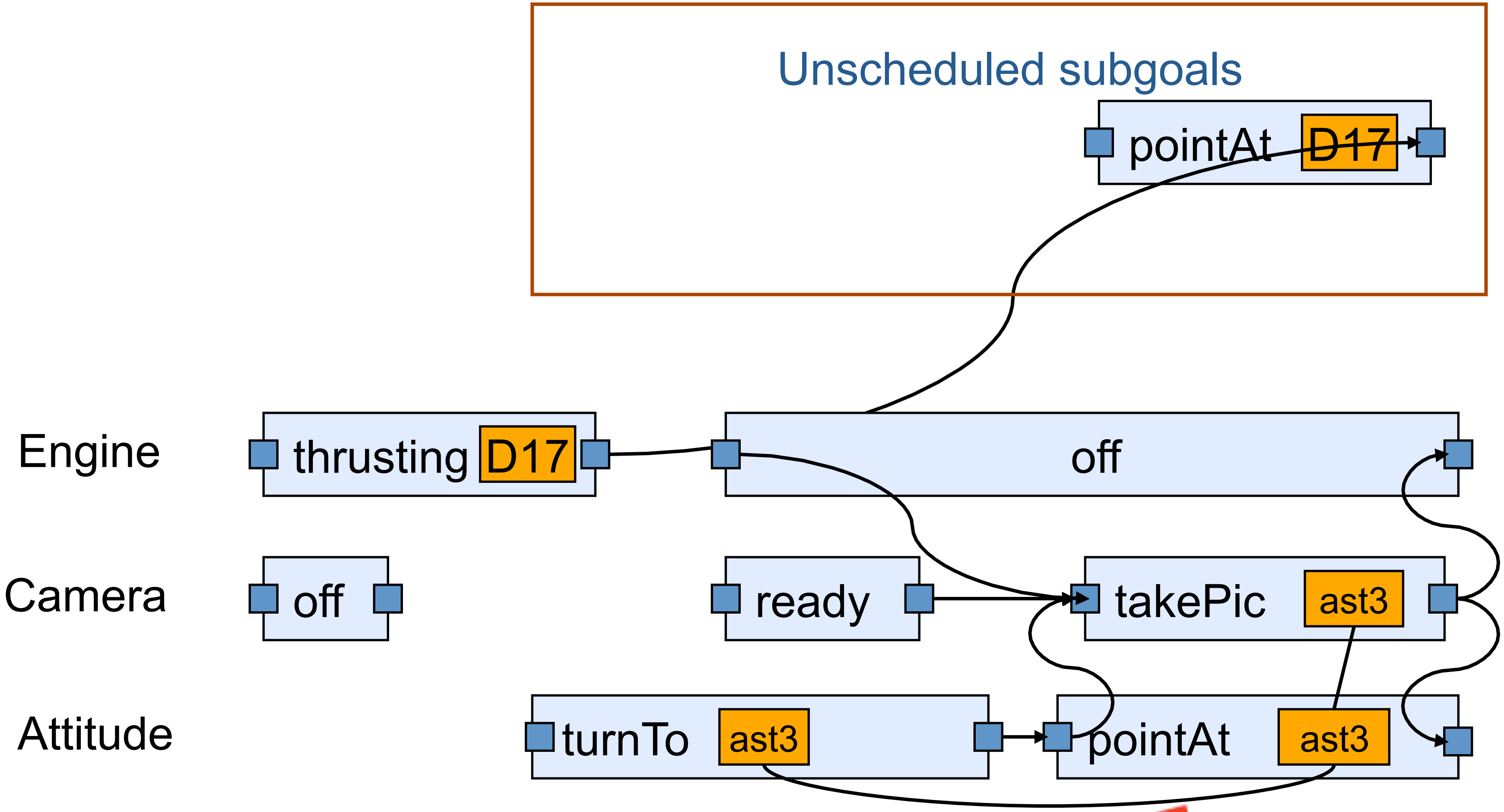
Alternate “change plan” option



Alternate "change plan" option

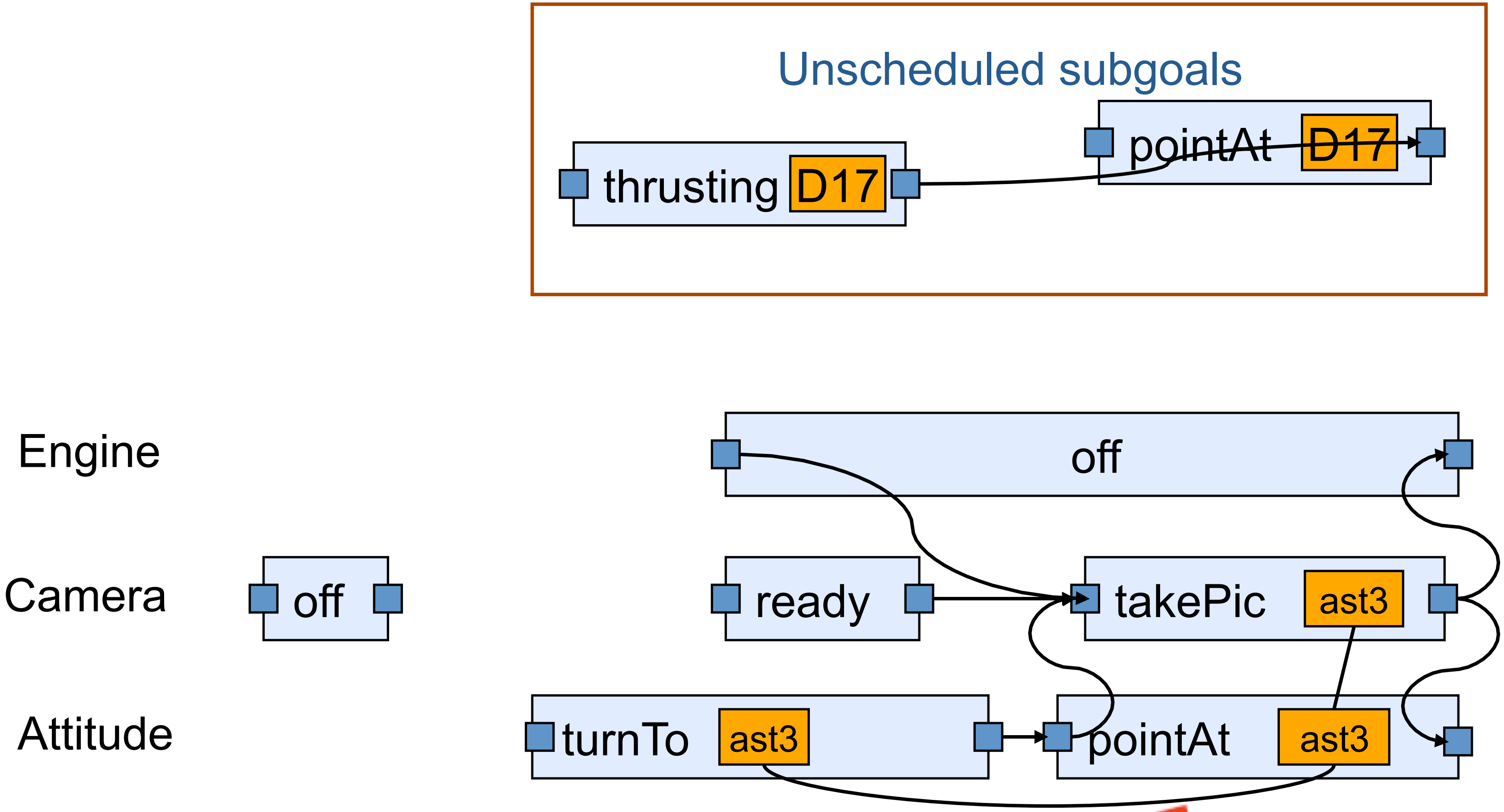


Alternate “change plan” option



REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Alternate "change plan" option



Interactive Planning

- Motivation for interactive planning
  - Human users can provide subjective evaluations
    - Impossible to encode all nuances and tradeoffs in model
    - Some tradeoffs are also “arbitrary”
  - Helps users understand and accept plans
    - Building plan interactively and incrementally helps understanding
- Elements of interactive planning
  - Interface for users to work on plan and interact with automation
  - User control over process
  - Range of automated reasoning services
    - Continuously operating services, e.g, active constraint enforcement
    - Requested services, e.g, resolve conflicts

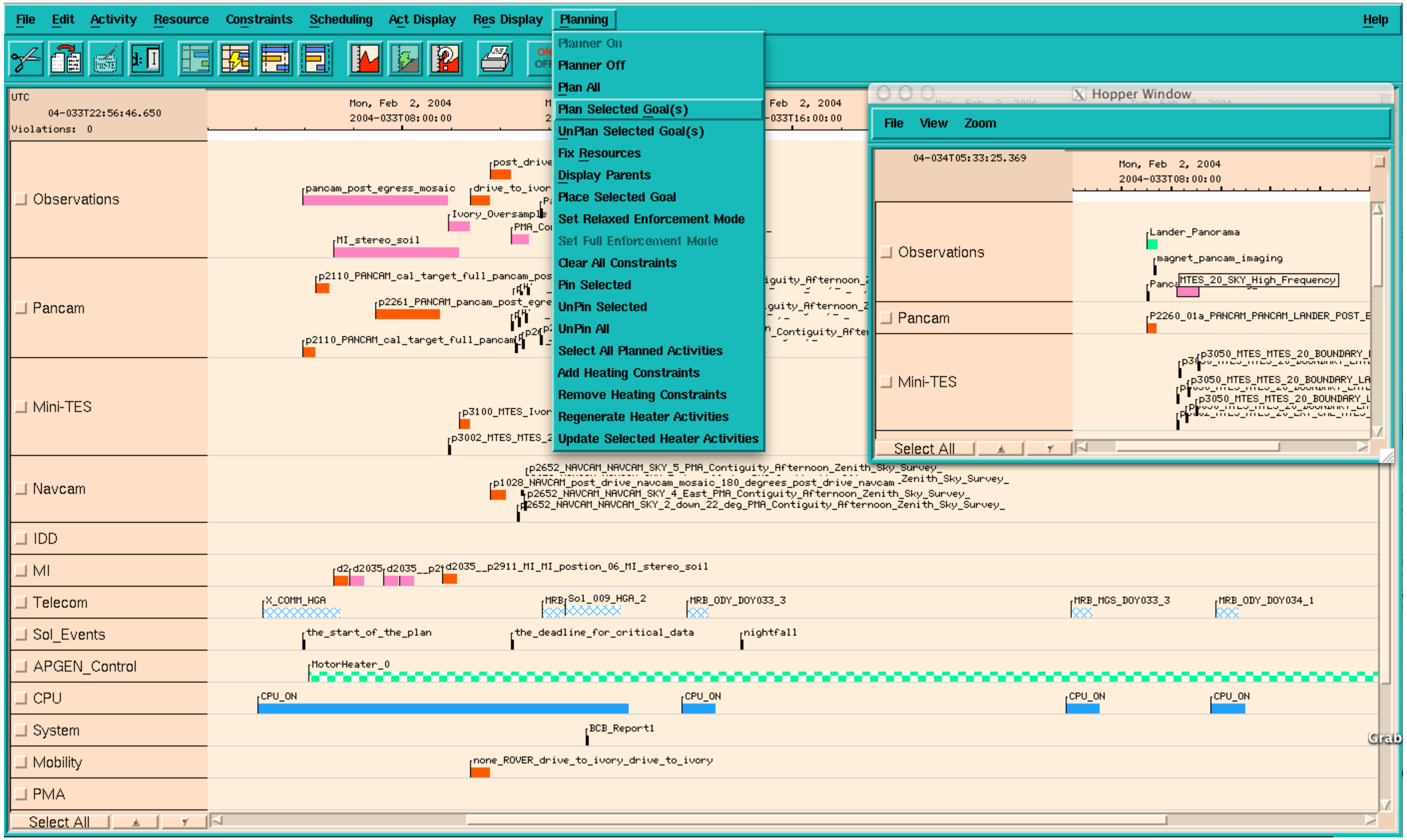
Back to MER

PLANNING IN ACTION

EUROPA in MAPGEN

- Full planning expressiveness
  – But, non-cotemporal subgoaling used sparingly
- Full plan optimization algorithm
  – But, rarely used in practice
- Limited planning used extensively
  – Users select goal but planner handles choices
- Constraint reasoning used extensively
  – Plan modifications
  – Time handling and scheduling

MAPGEN interface

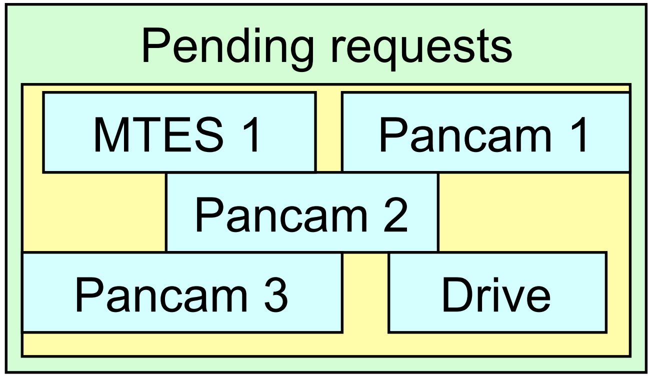


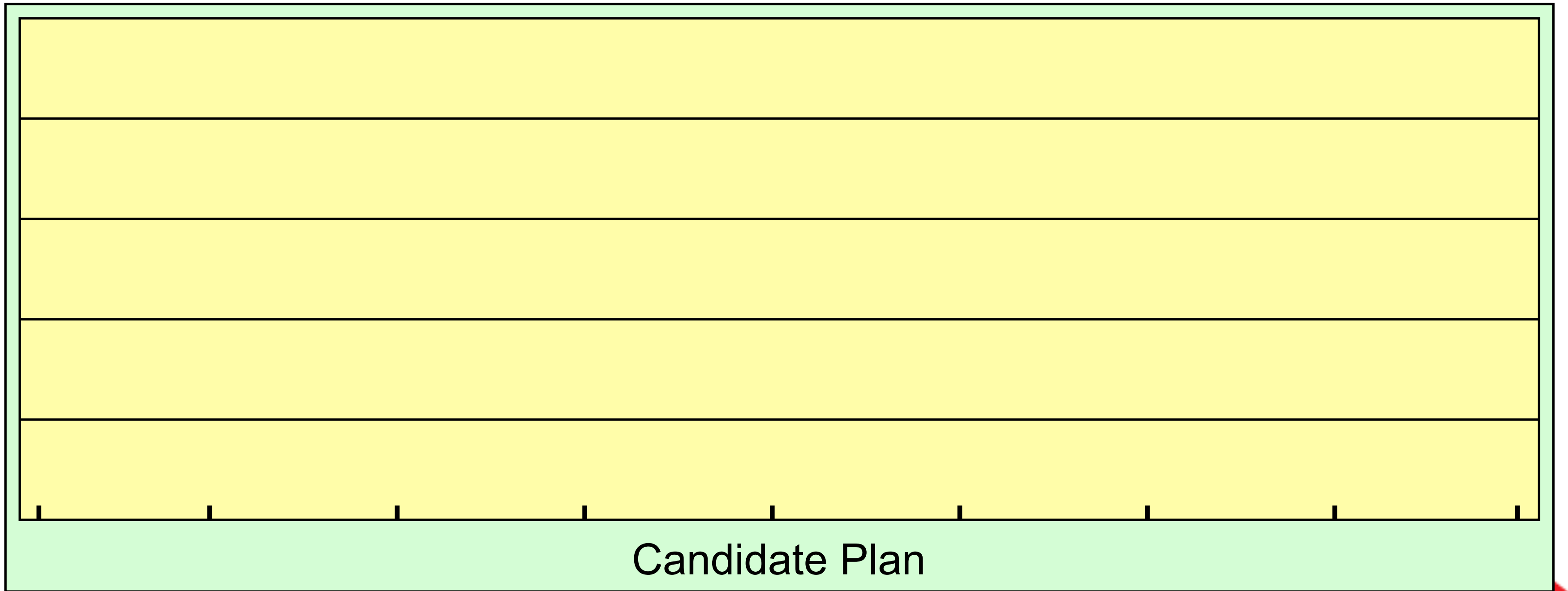
Usage scenario

- Repeat until plan is ready:
  - Add more requests to the plan
    - Different users used different approaches
  - Evaluate plan
    - Resource usage
    - Subjective preferences
  - Modify plan as needed to
    - Satisfy resource limitations
    - Satisfy rules that are not in domain model
    - Respect preferences to the extent possible
- In case of new information about rover:
  - Make necessary changes to plan
- Typical plan: 100 “goals” and 3500 activities

Example 1: Simple plan

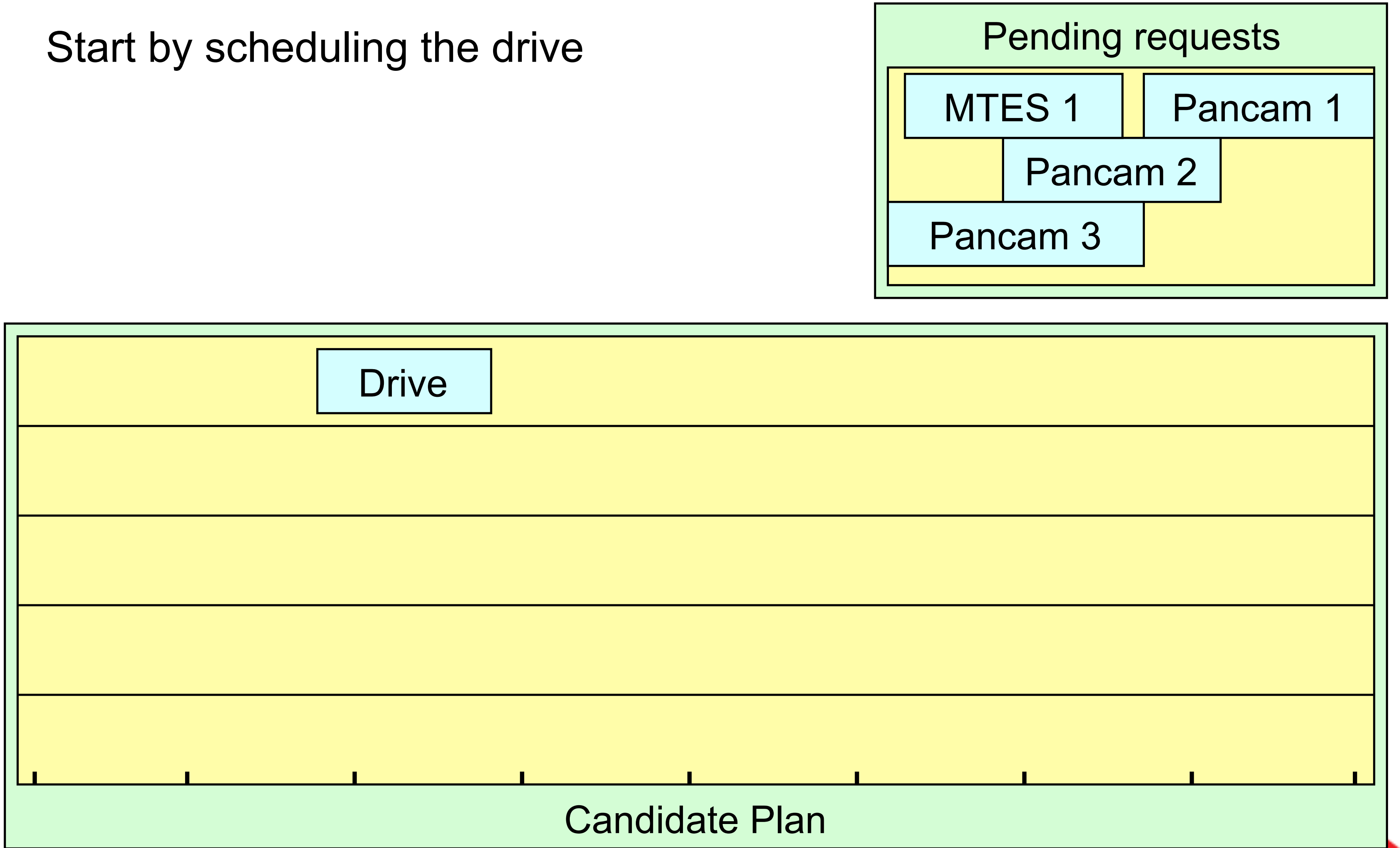
Drive must be early
Pancam 1 is before drive
Other activities after drive
Drive most important, then PC1, PC3, MTES1 and PC2





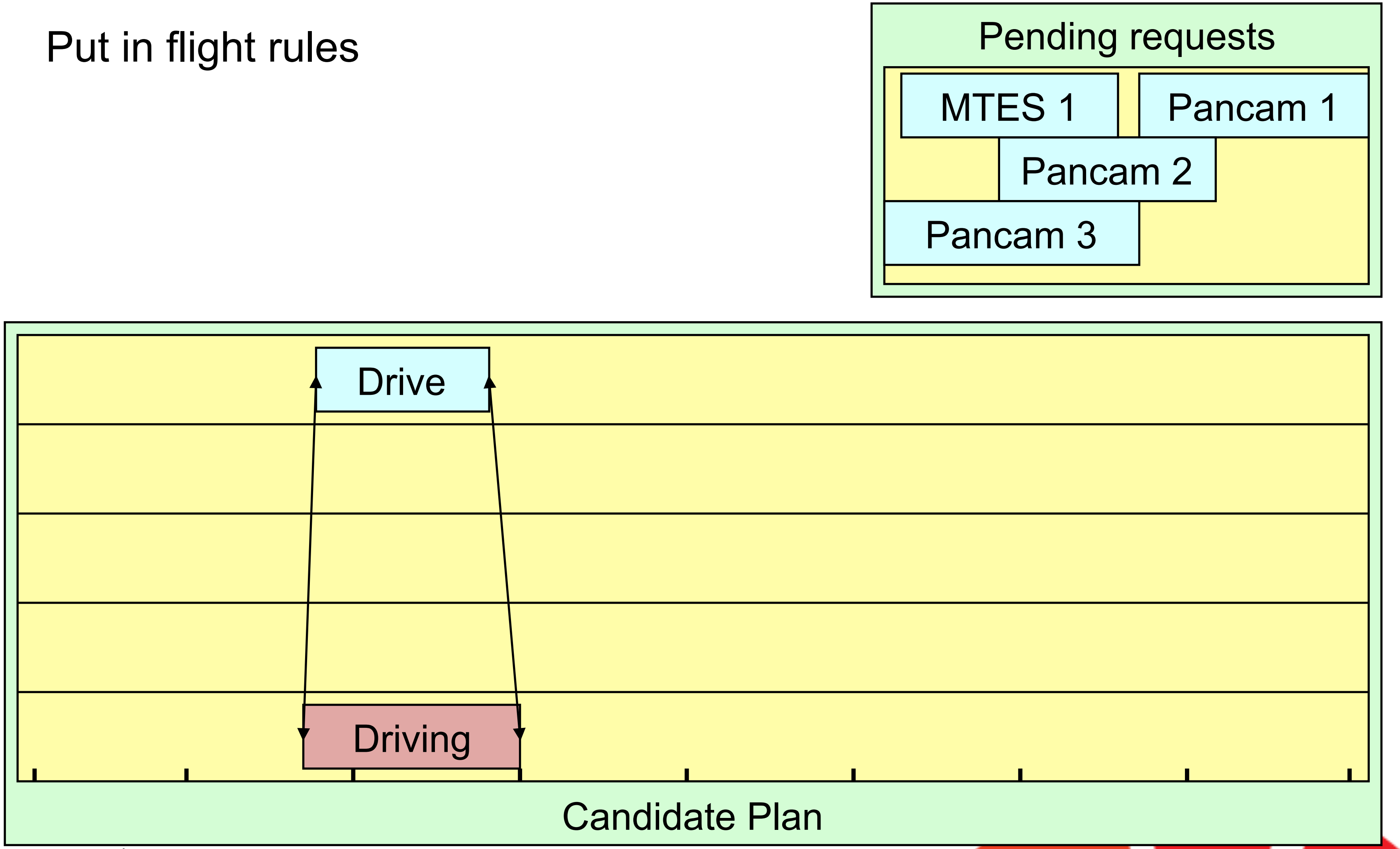
Example 1: Simple plan

Start by scheduling the drive



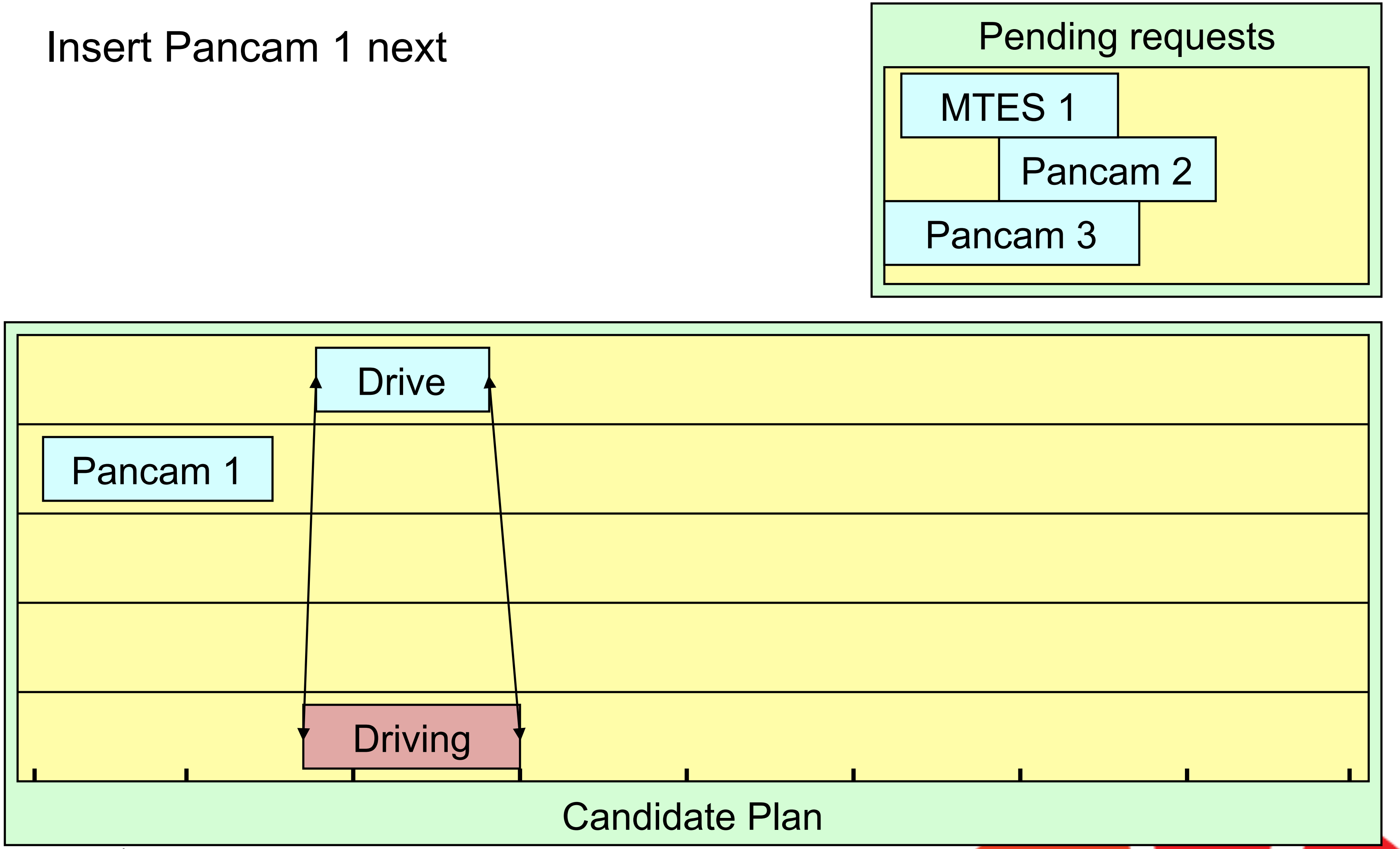
Example 1: Simple plan

Put in flight rules



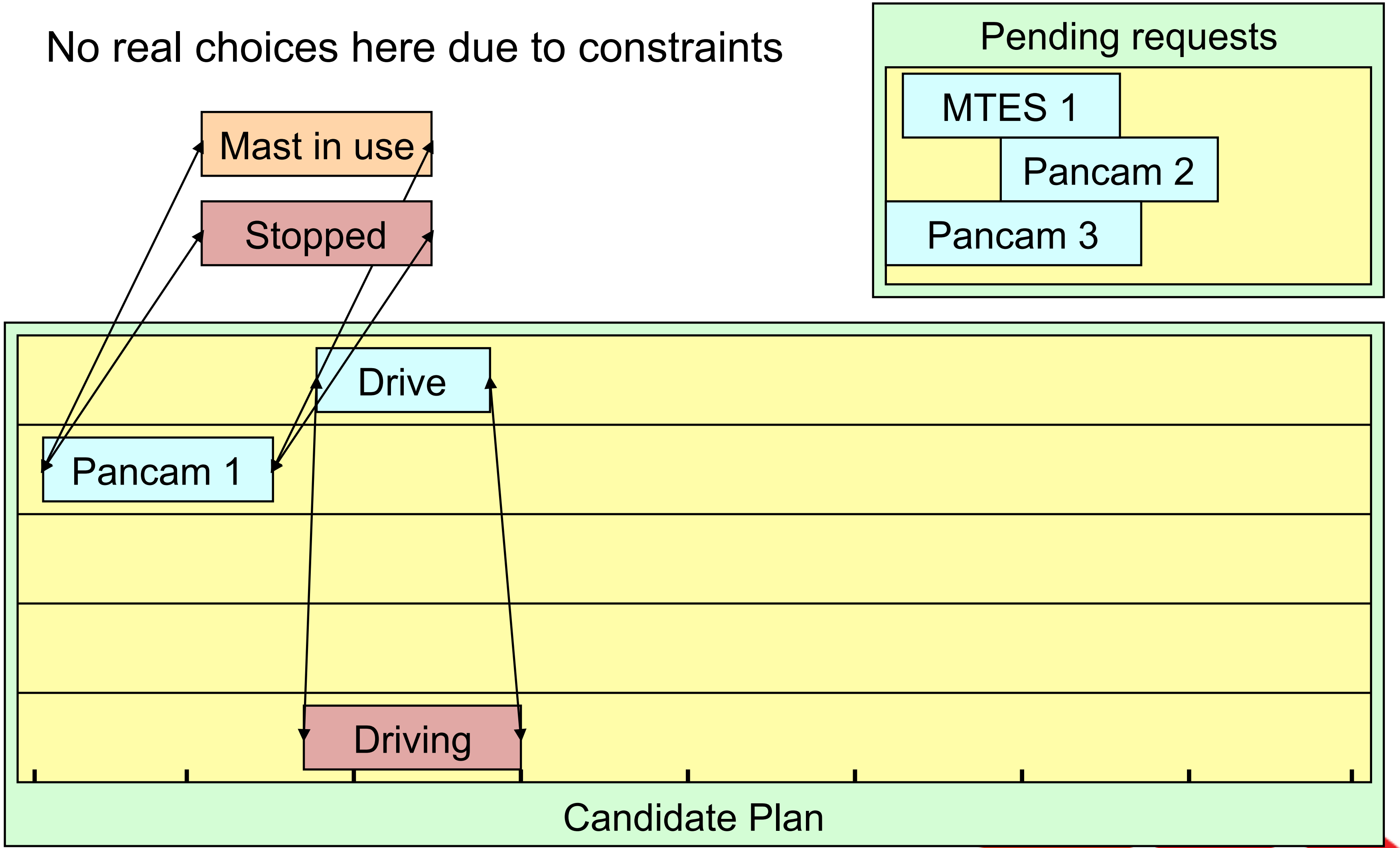
Example 1: Simple plan

Insert Pancam 1 next



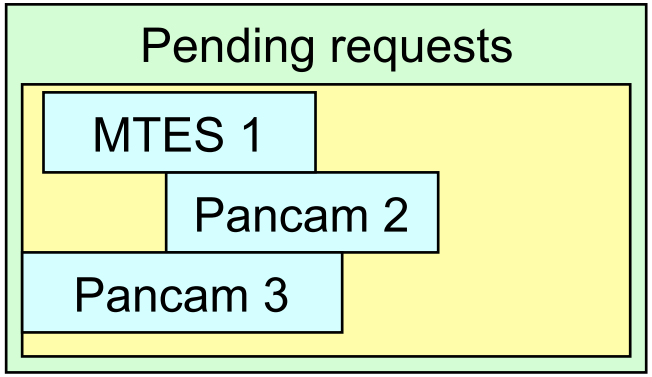
Example 1: Simple plan

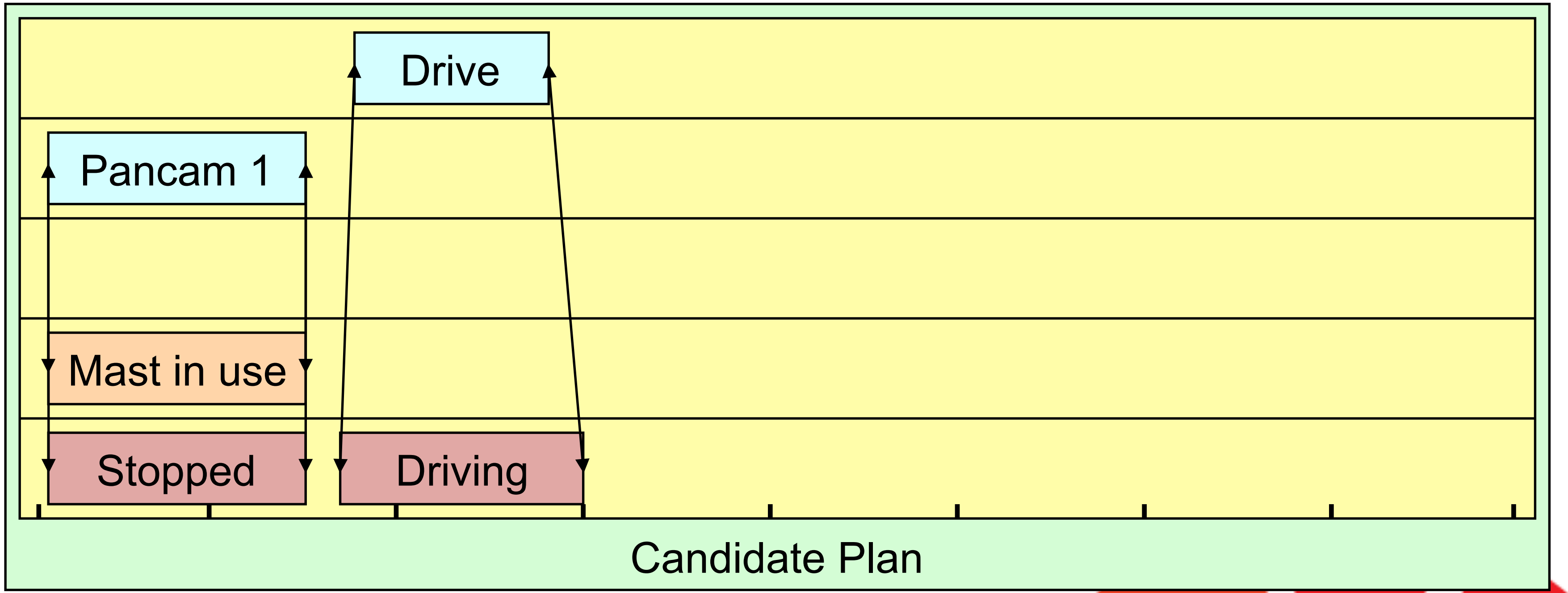
No real choices here due to constraints



Example 1: Simple plan

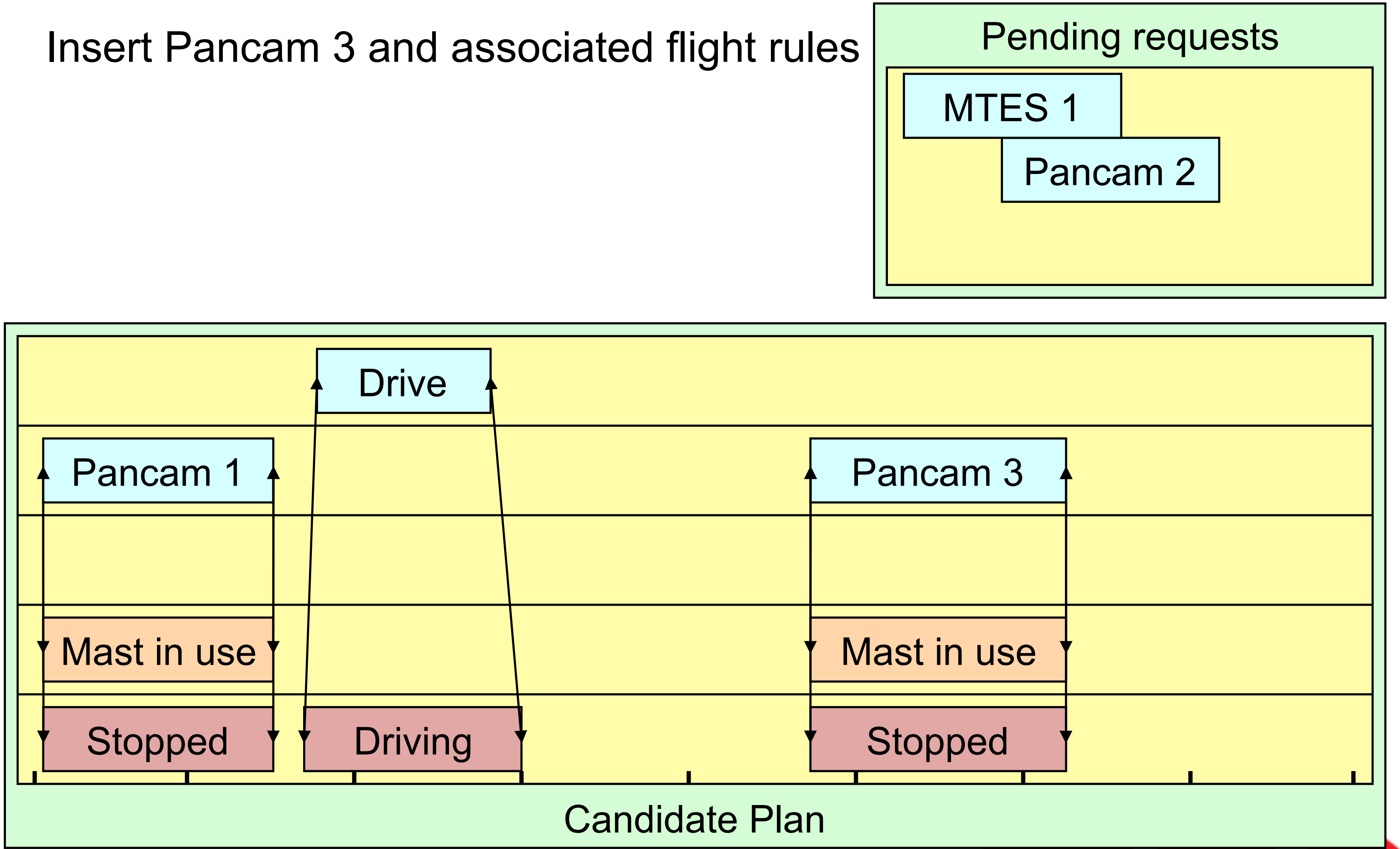
Enforce flight rules





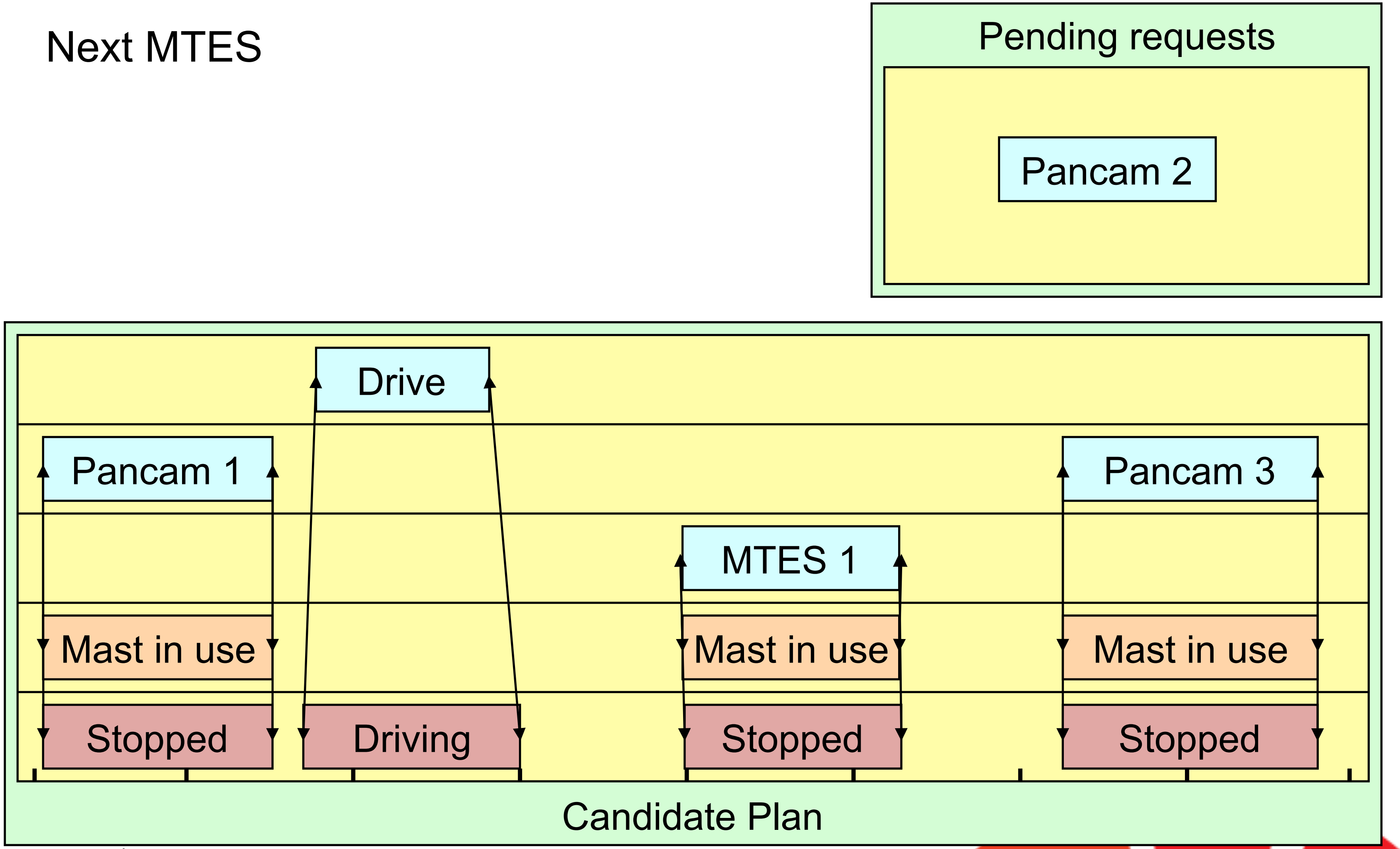
Example 1: Simple plan

Insert Pancam 3 and associated flight rules



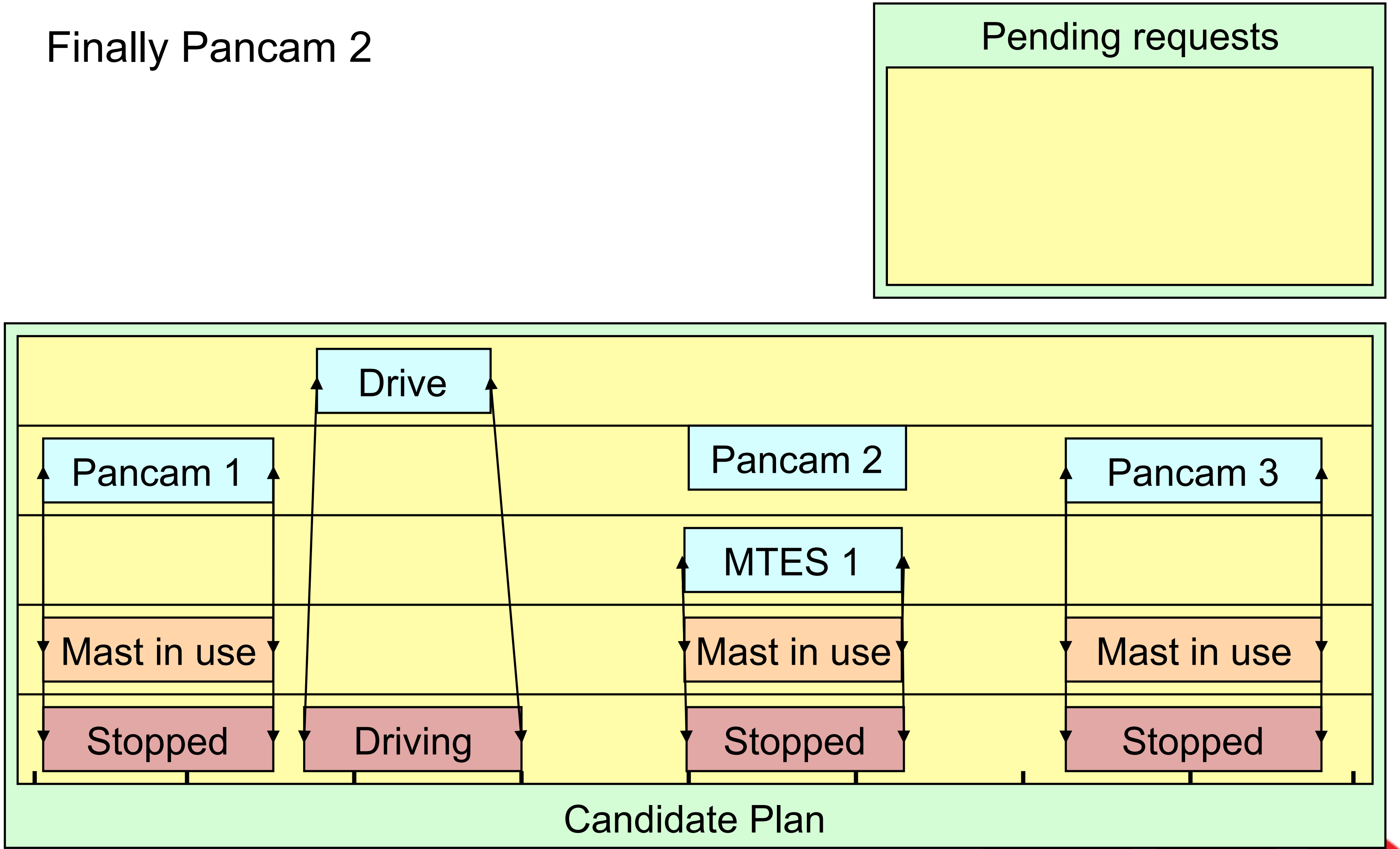
Example 1: Simple plan

Next MTES



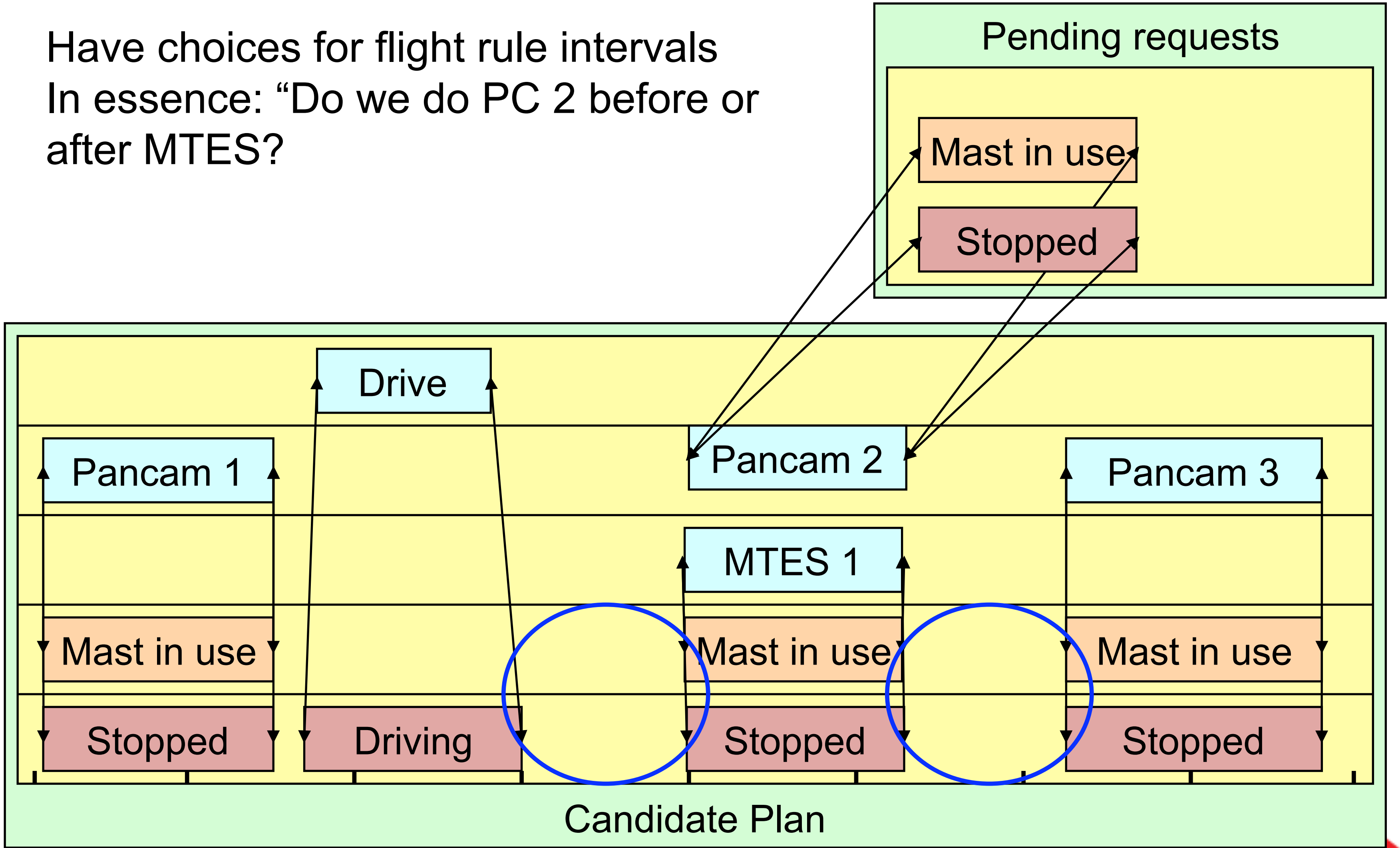
Example 1: Simple plan

Finally Pancam 2



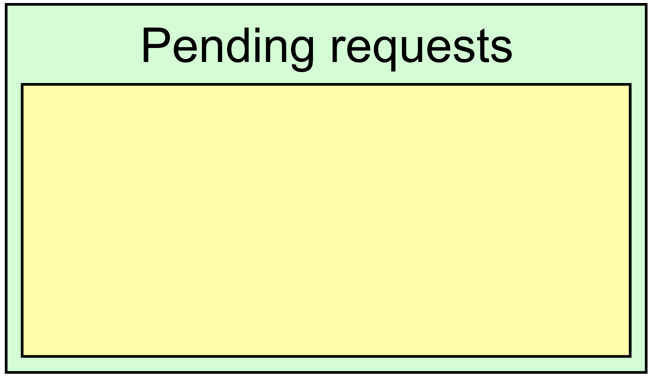
Example 1: Simple plan

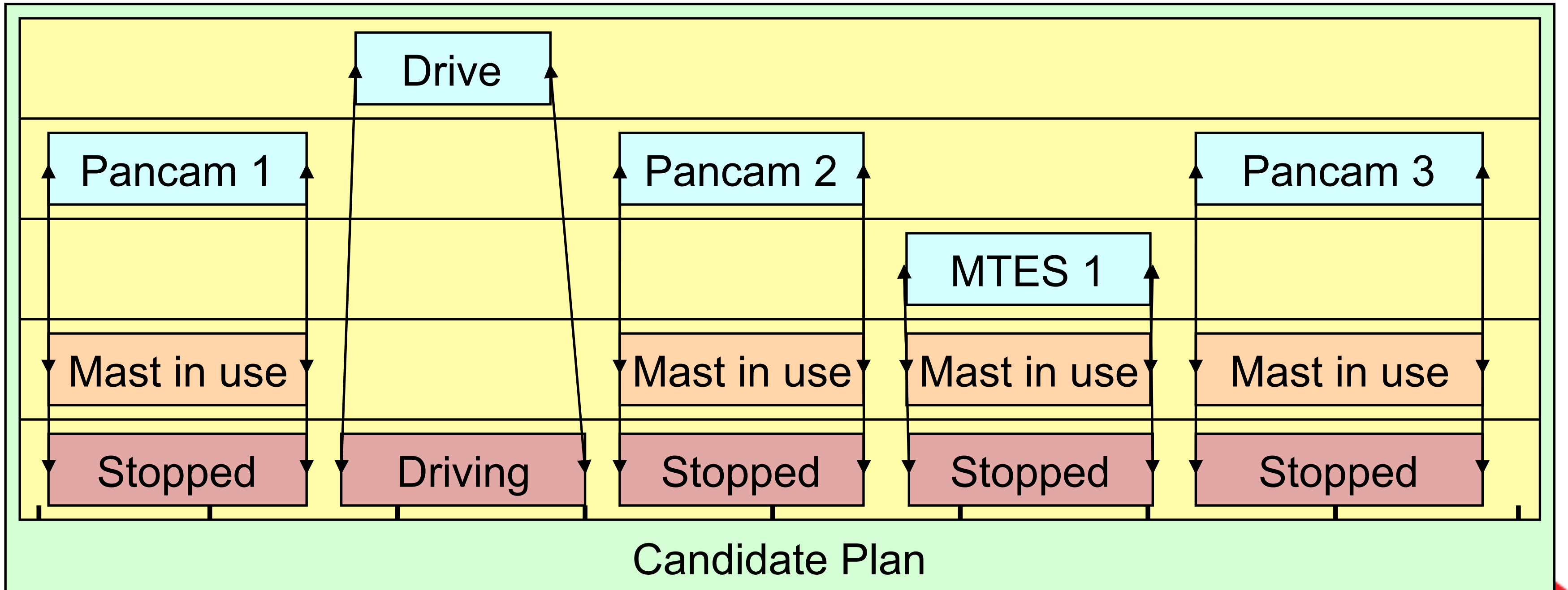
Have choices for flight rule intervals
In essence: "Do we do PC 2 before or after MTES?



Example 1: Simple plan

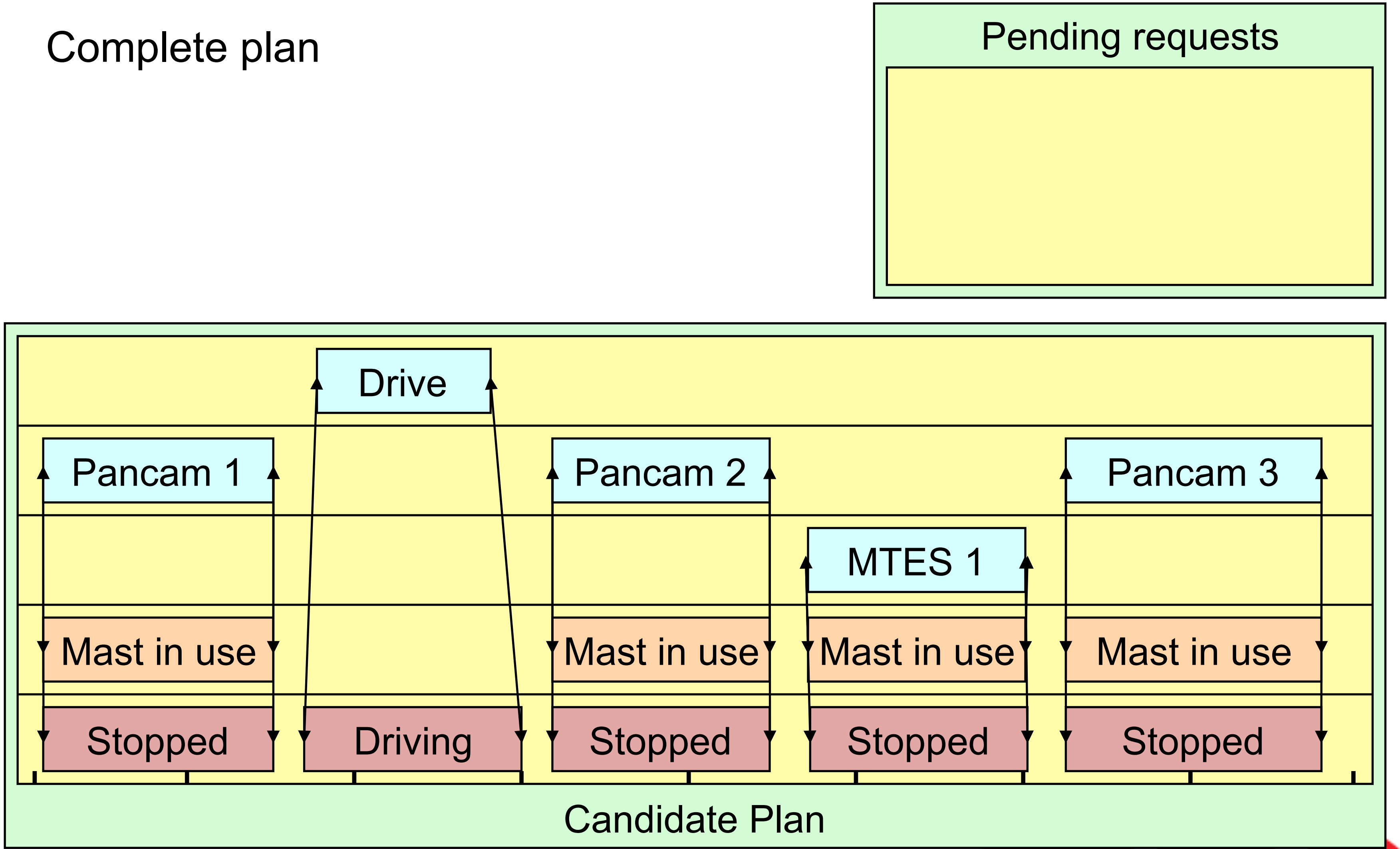
Put it after the MTES





Example 1: Simple plan

Complete plan

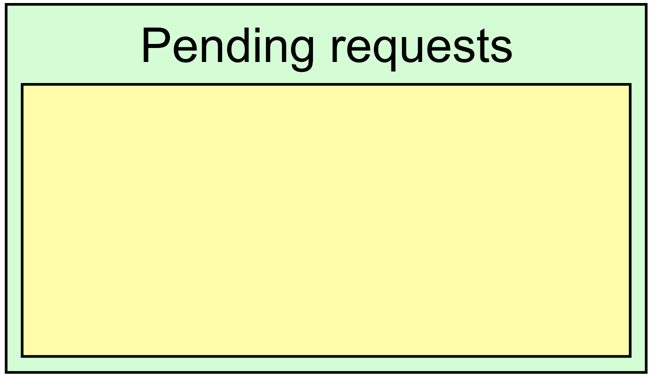


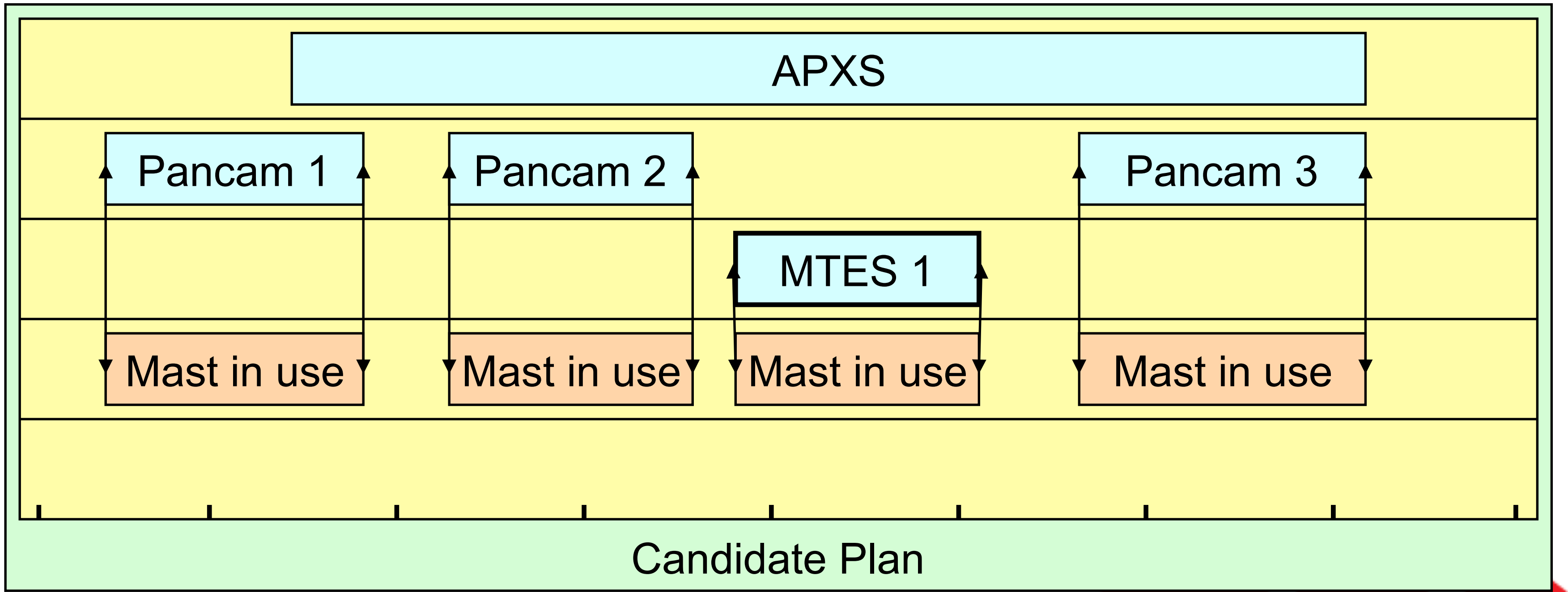
Working with plans

- Unexpected results
  - Plan not as good as it should be
  - Plan not “the one” expected
- Provided approaches
  - Moving activities around
  - Unplan and replan selected activities
  - Relaxed mode
  - Modify constraints and go back to planner

Example 2: Simple move

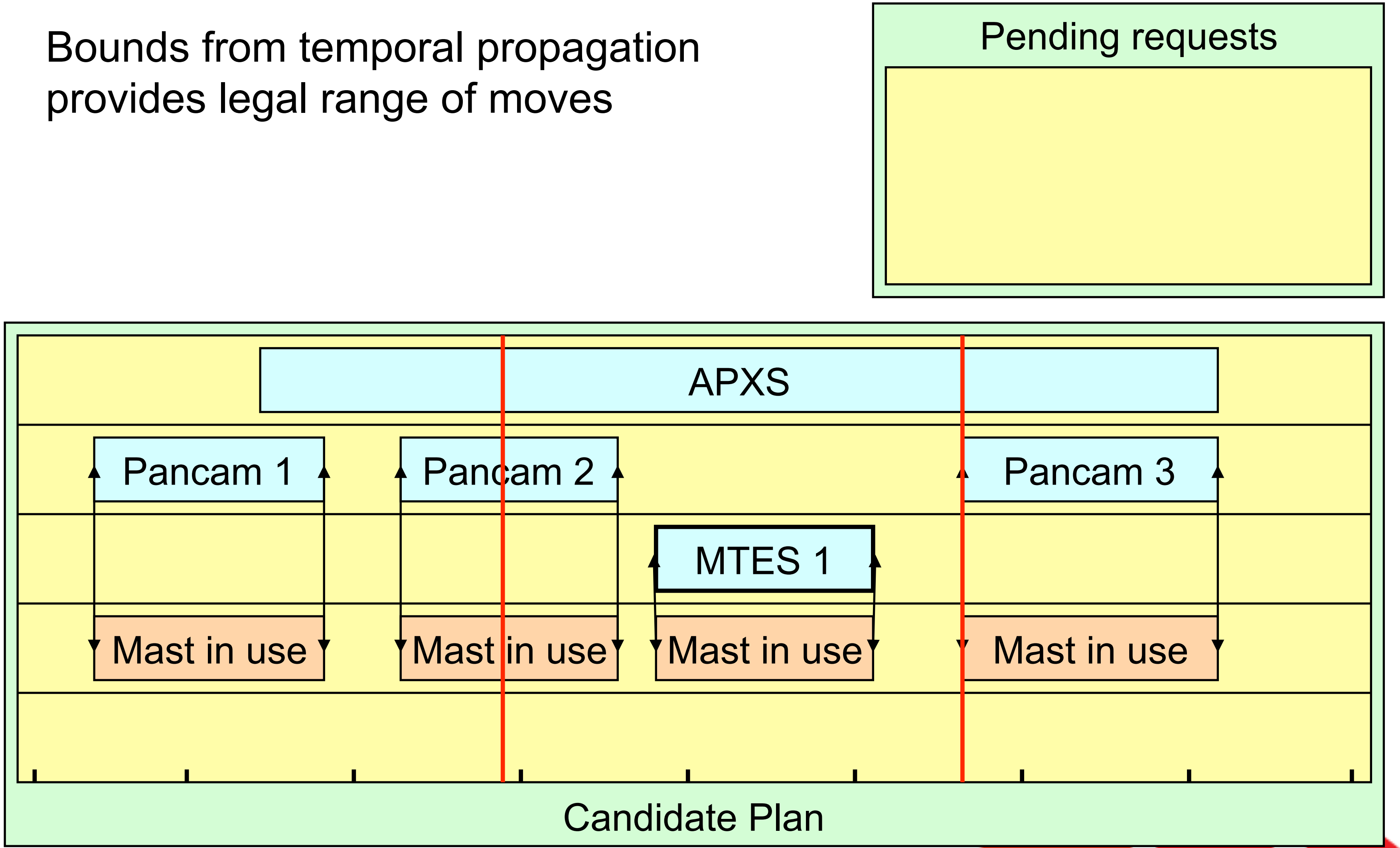
User selects and moves **MTES 1**





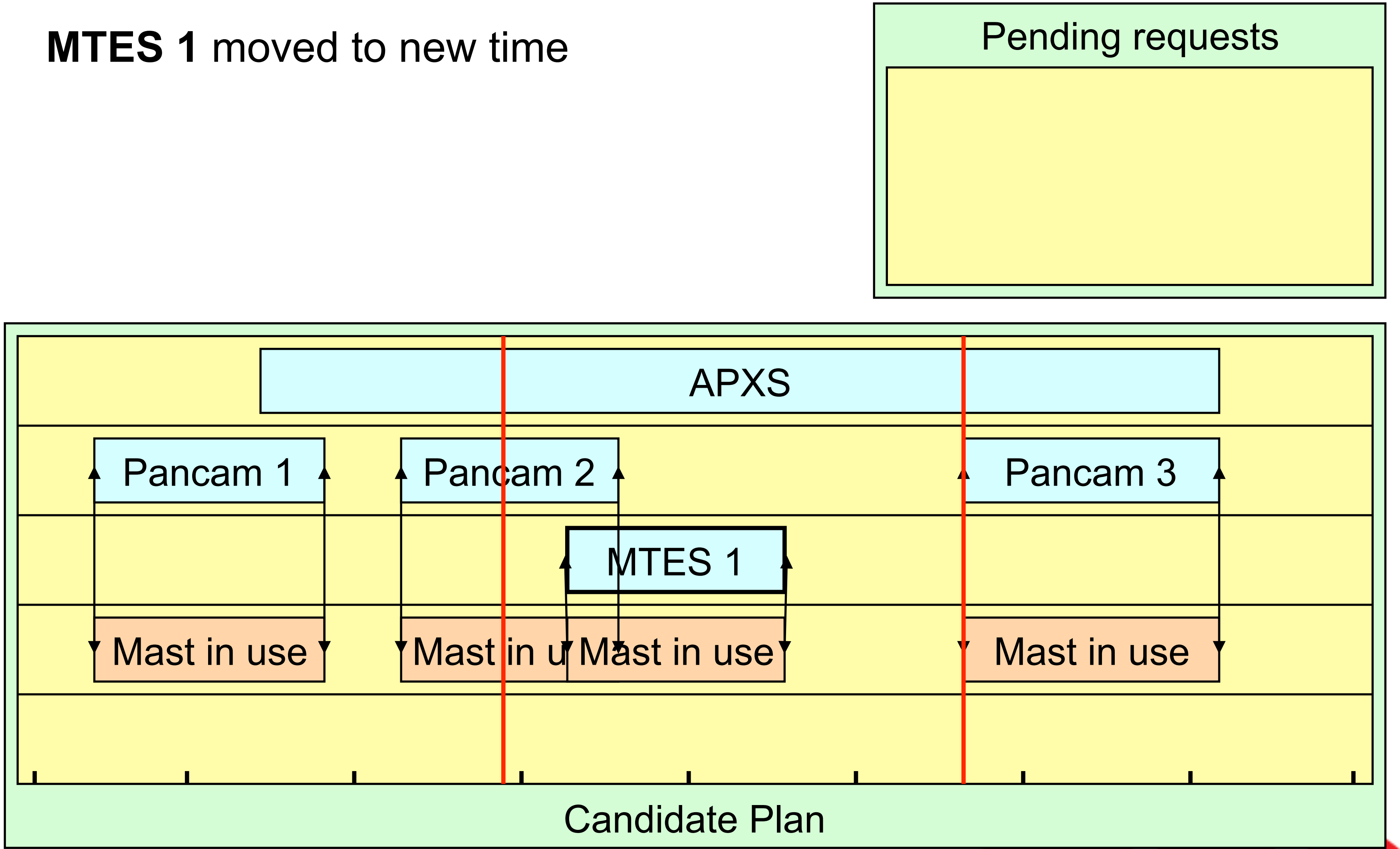
Example 2: Simple move

Bounds from temporal propagation provides legal range of moves



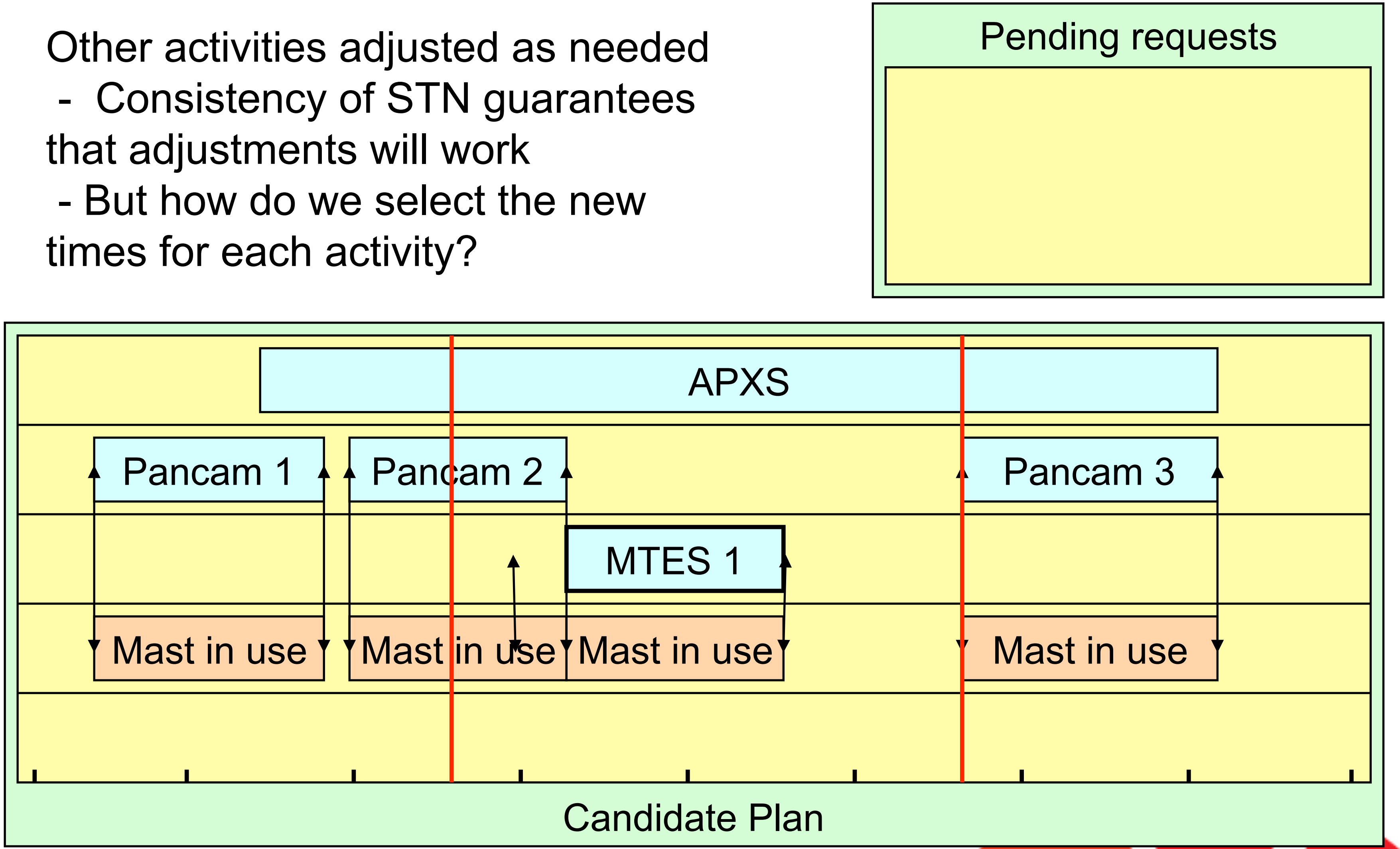
Example 2: Simple move

MTES 1 moved to new time



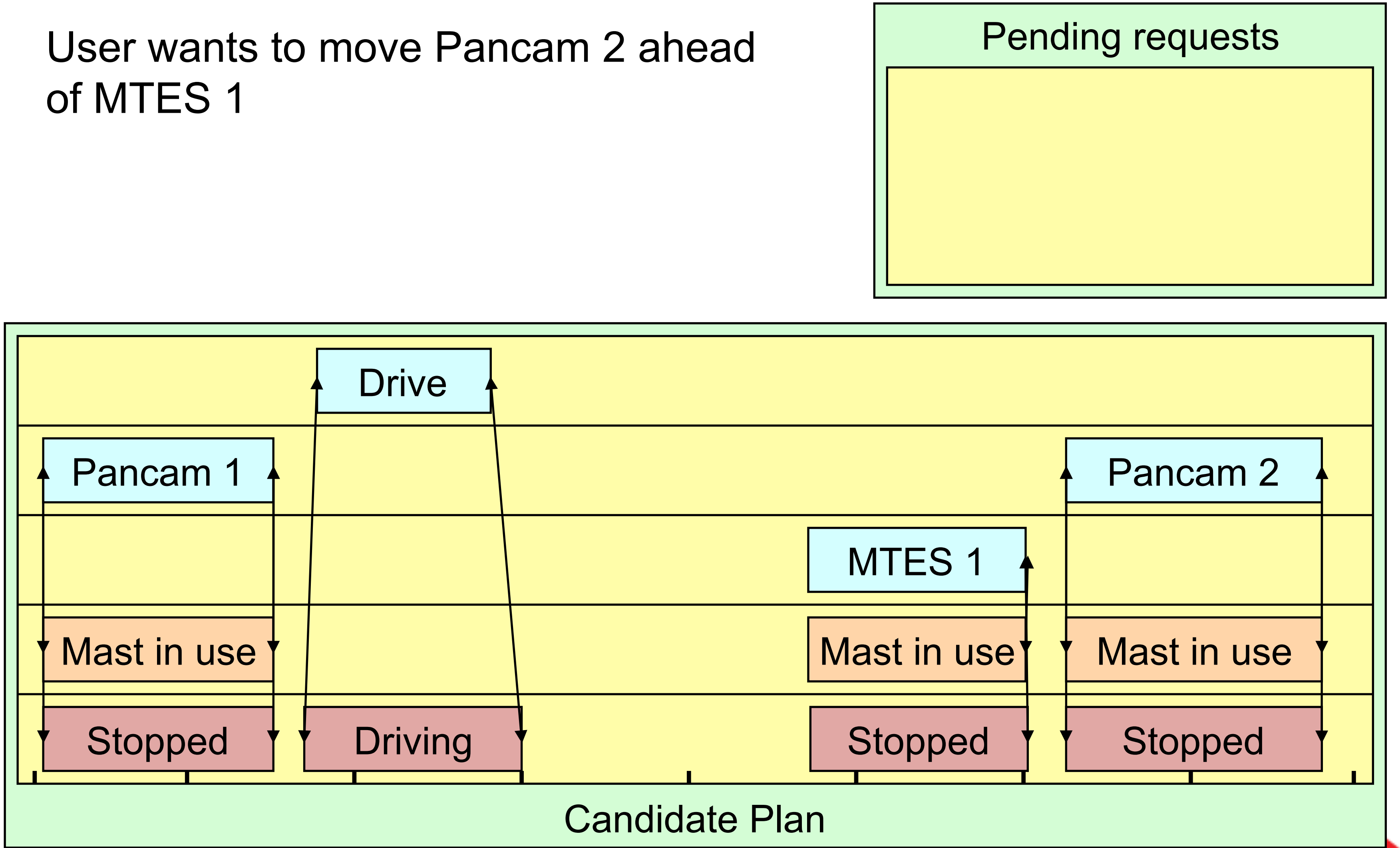
Example 2: Simple move

Other activities adjusted as needed
 - Consistency of STN guarantees that adjustments will work
 - But how do we select the new times for each activity?



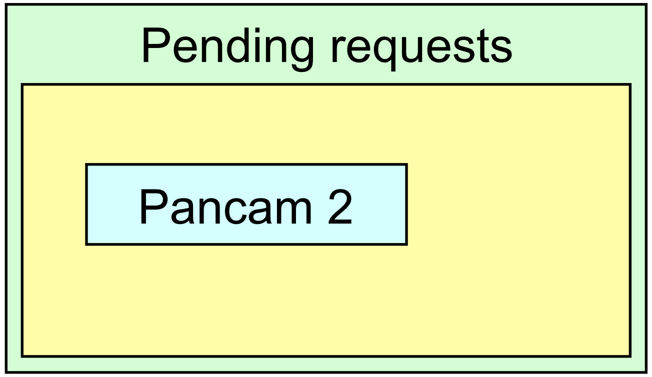
Example 3: Unplan + Place

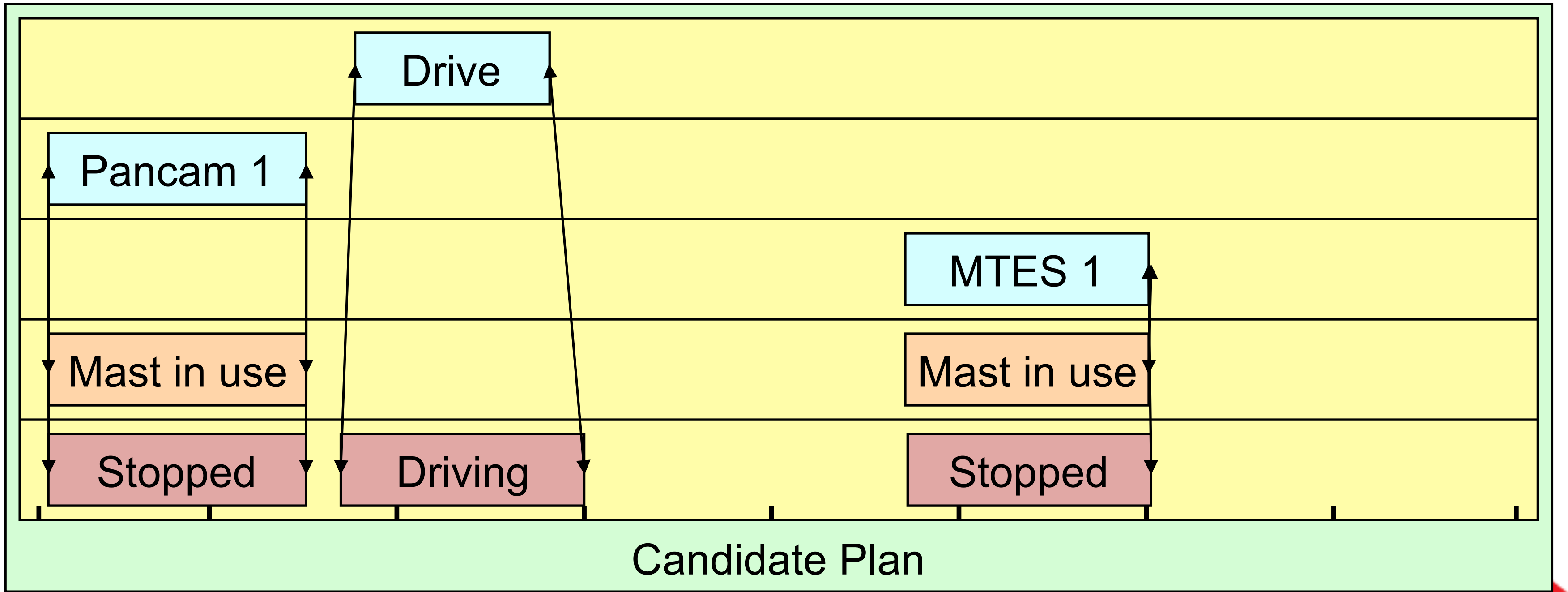
User wants to move Pancam 2 ahead of MTES 1



Example 3: Unplan + Place

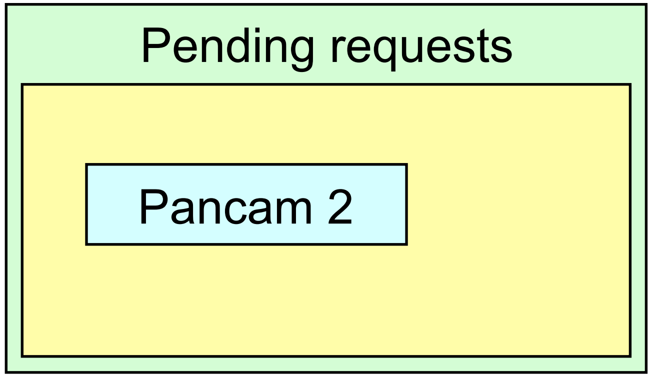
Unplan Pancam 2

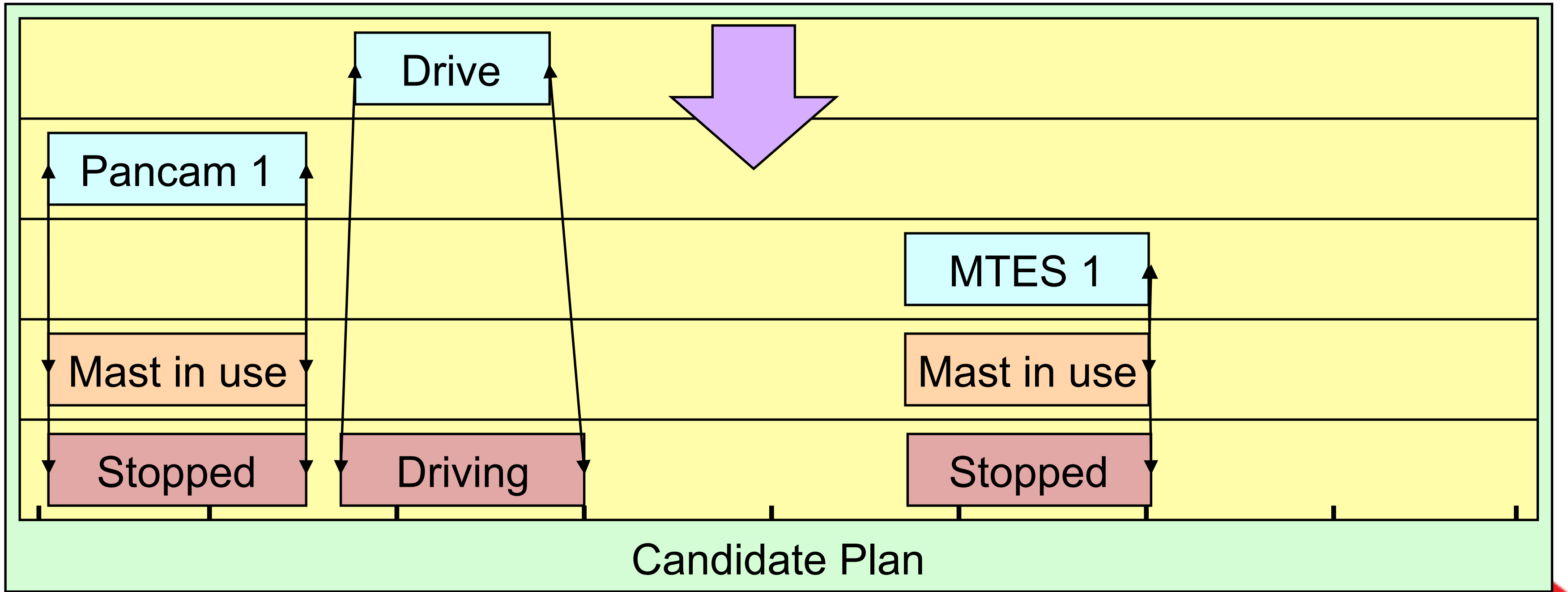




Example 3: Unplan + Place

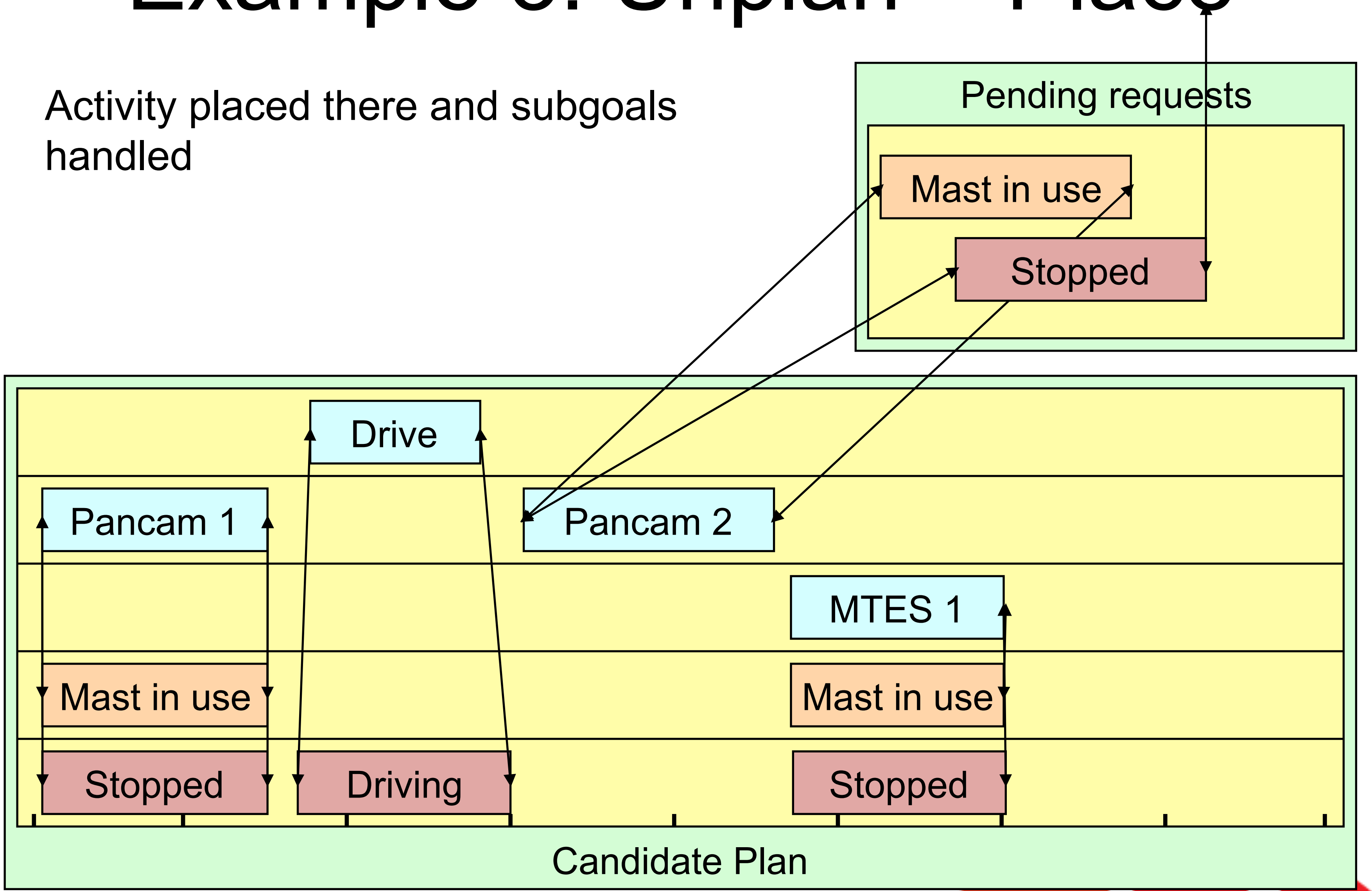
Select Pancam 2 and "place it" by clicking on desired part of plan





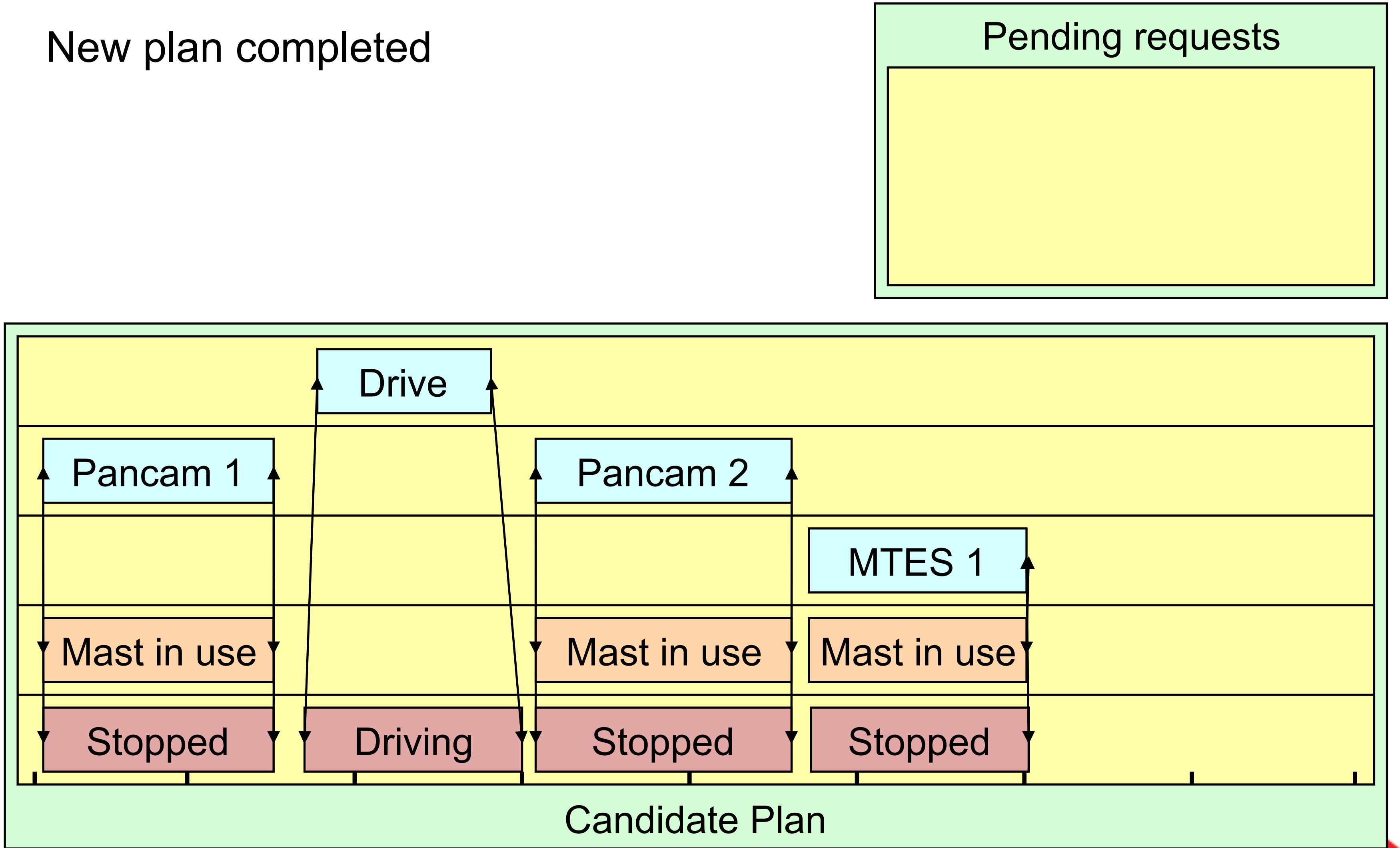
Example 3: Unplan + Place

Activity placed there and subgoals handled



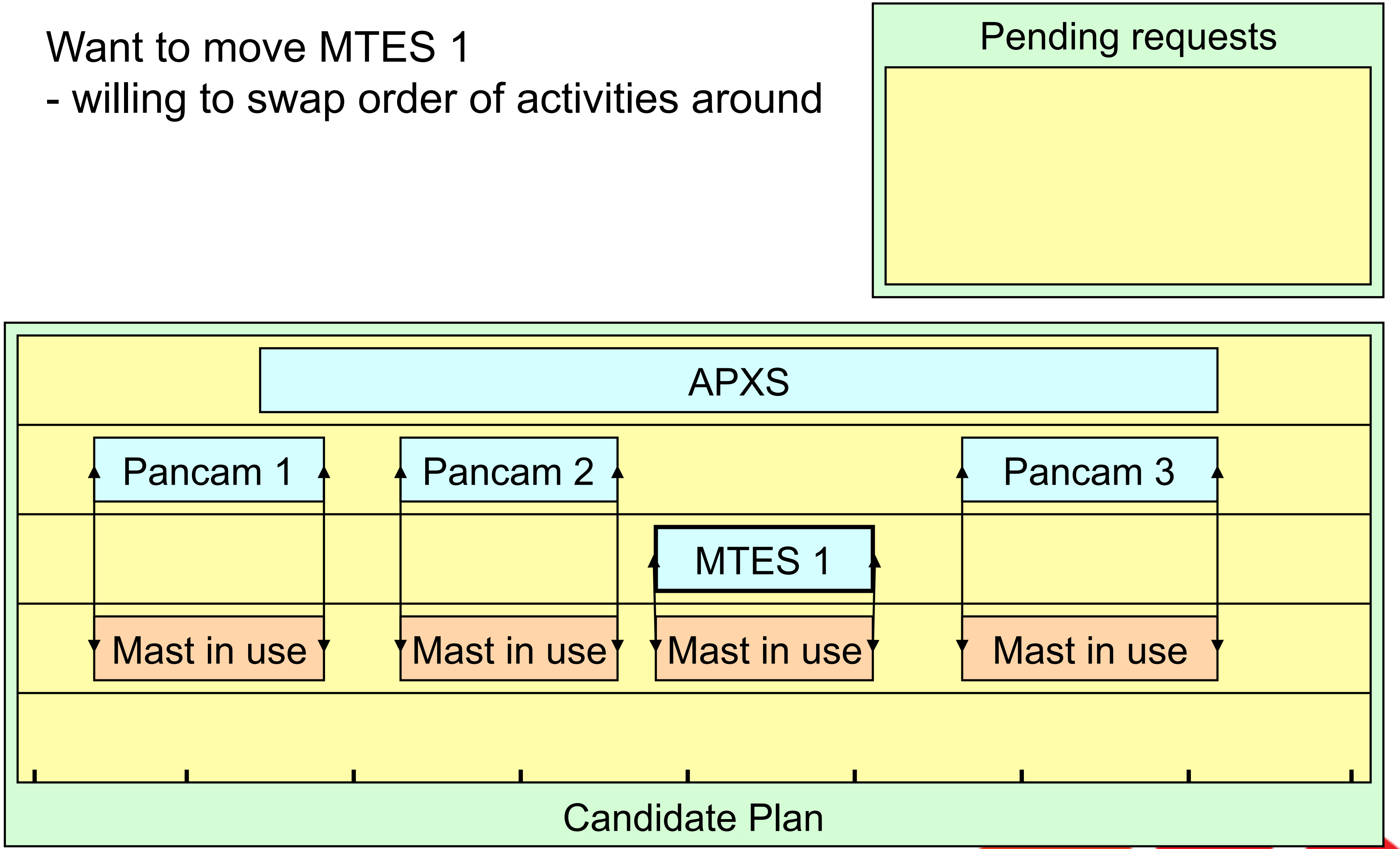
Example 3: Unplan + Place

New plan completed



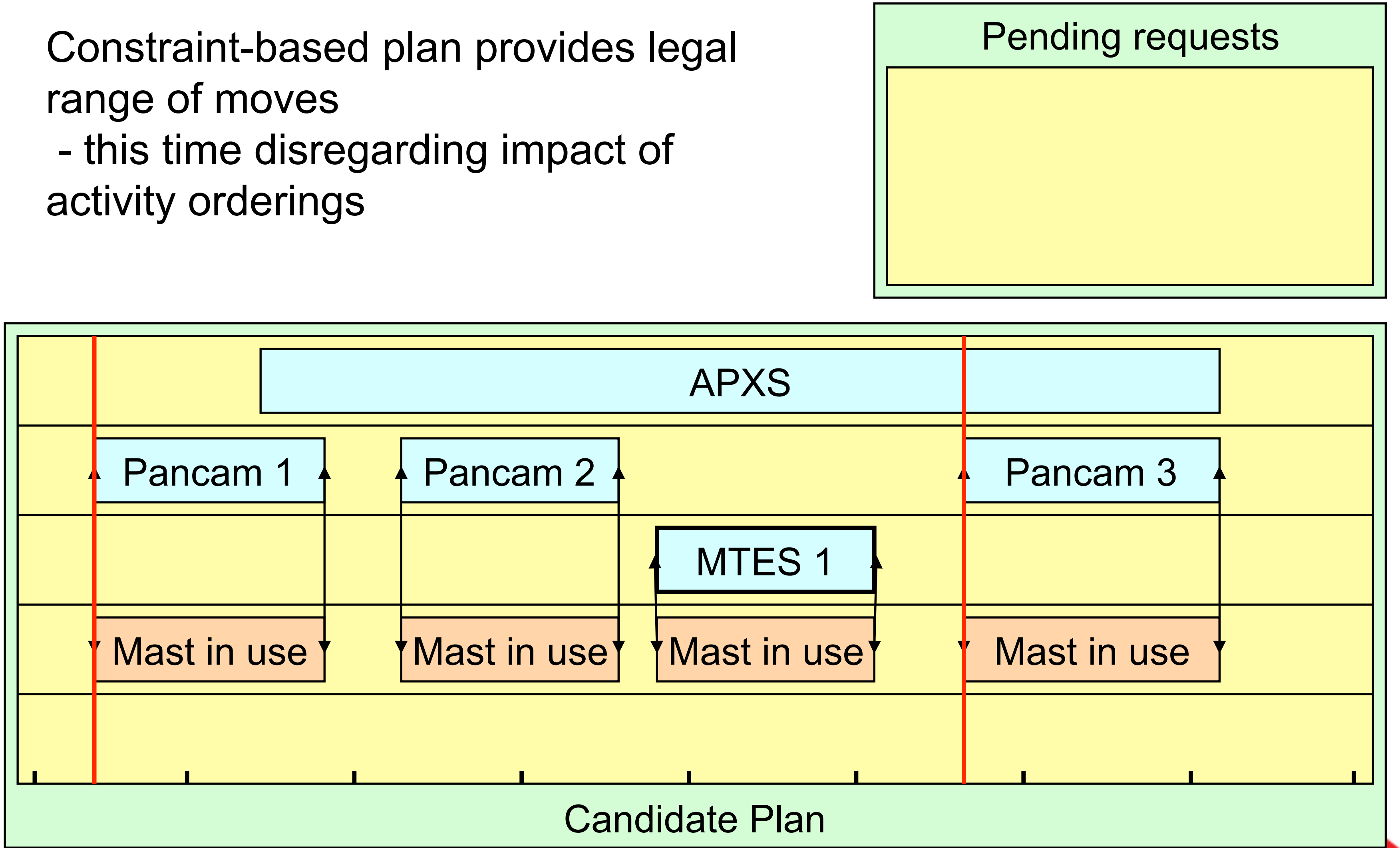
Example 4 - “Super move”

Want to move MTES 1
- willing to swap order of activities around



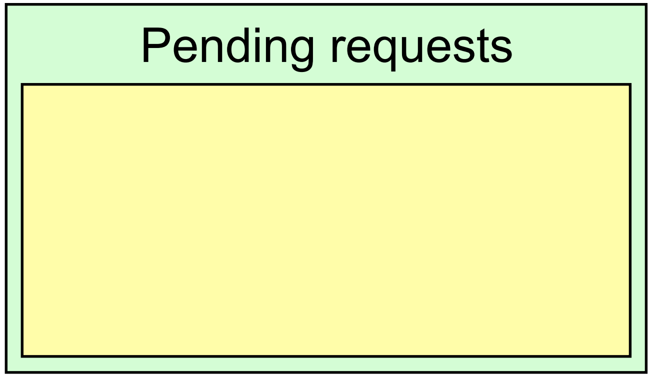
Example 4 - “Super move”

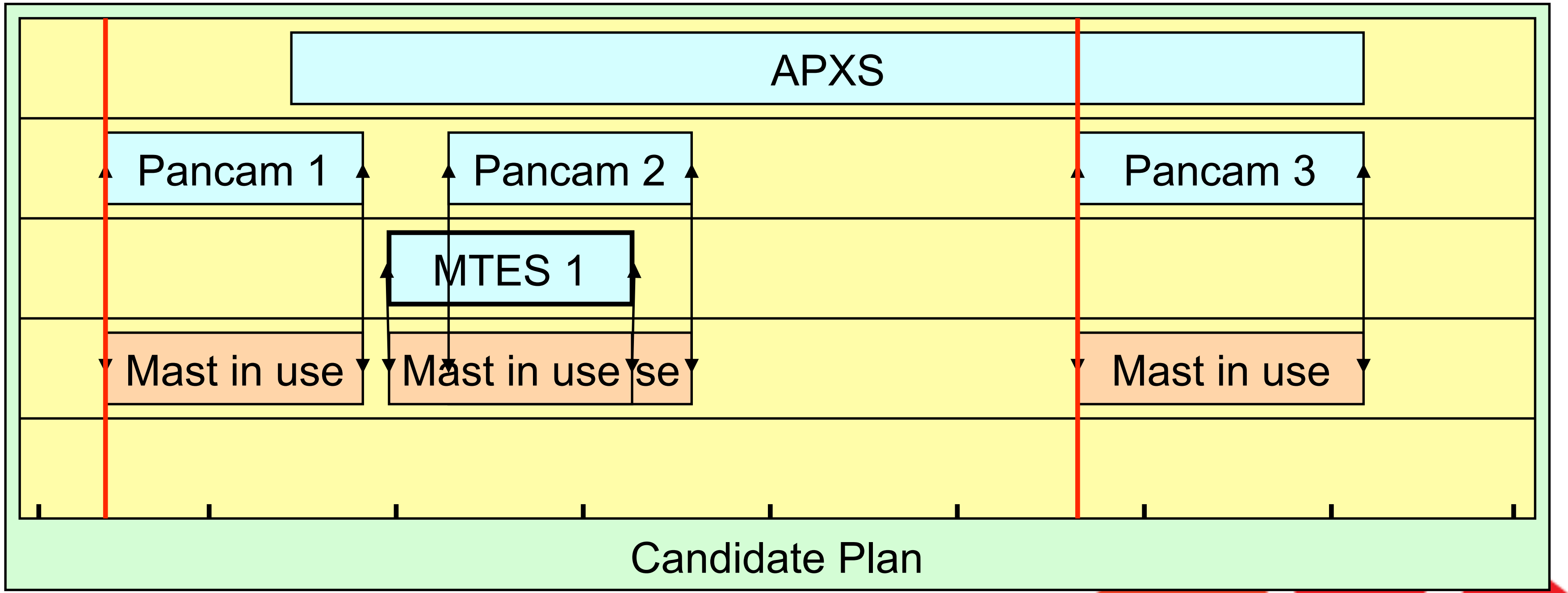
Constraint-based plan provides legal range of moves
 - this time disregarding impact of activity orderings



Example 4 - “Super move”

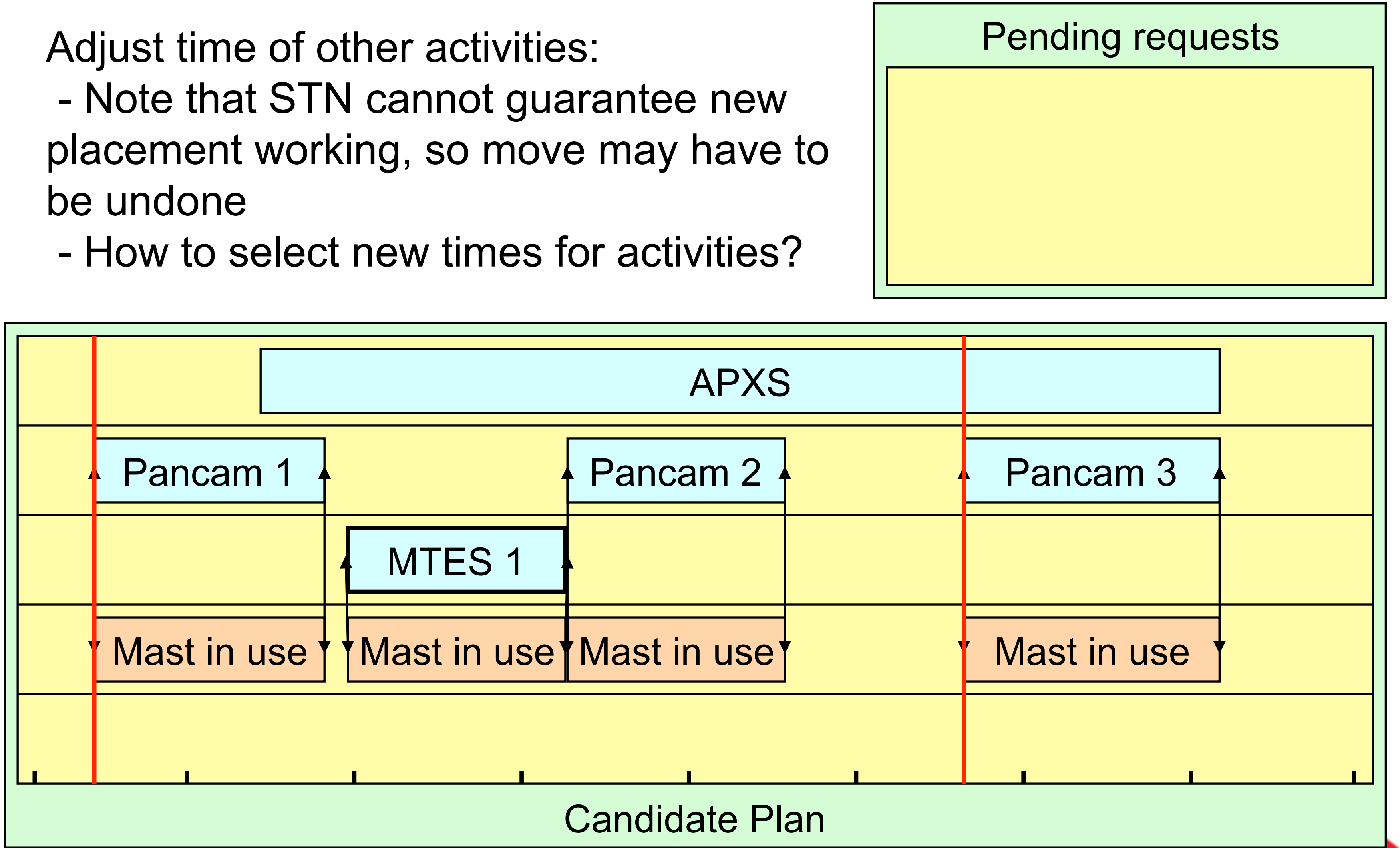
Move MTES 1 to new time





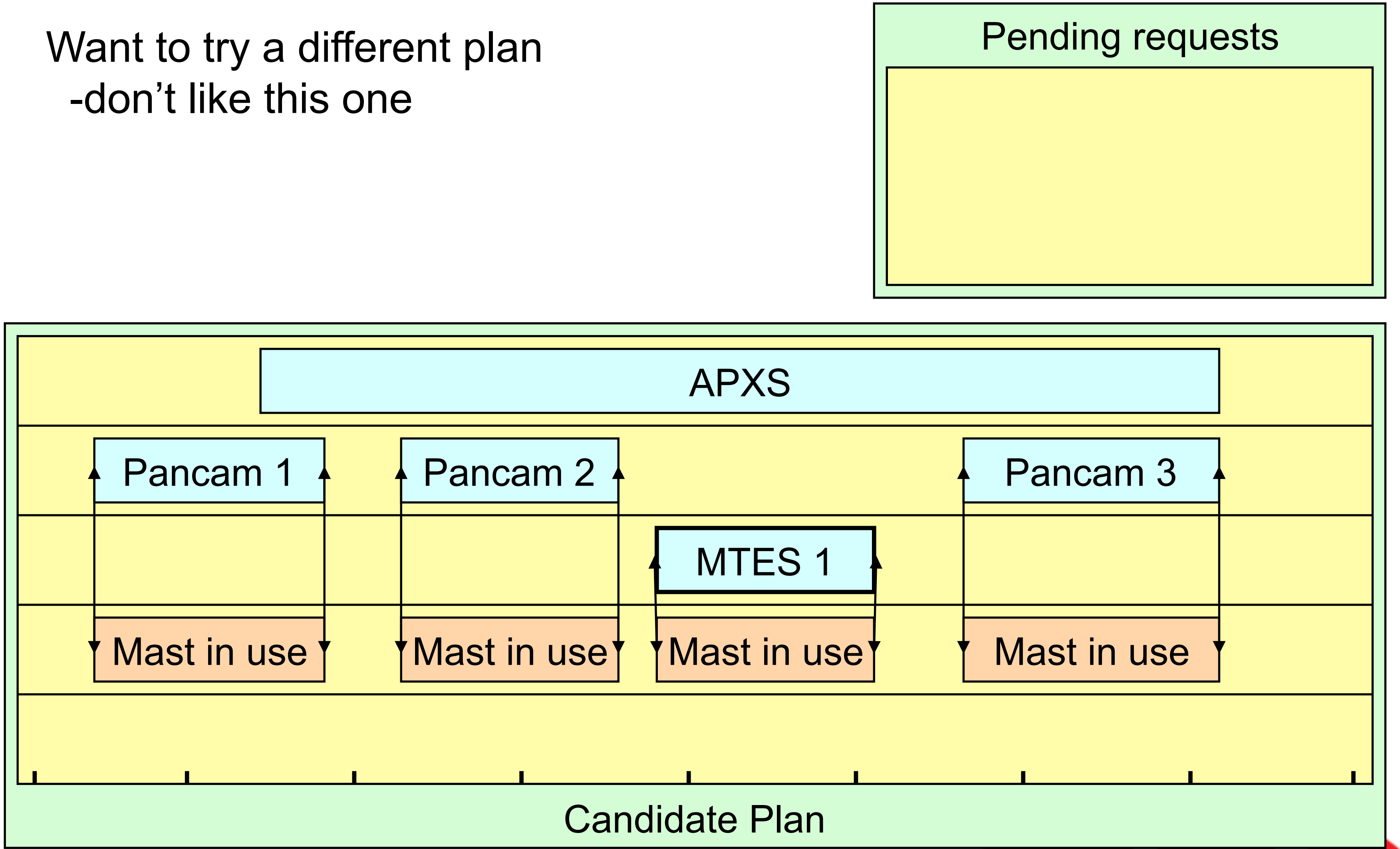
Example 4 - “Super move”

Adjust time of other activities:
- Note that STN cannot guarantee new placement working, so move may have to be undone
- How to select new times for activities?



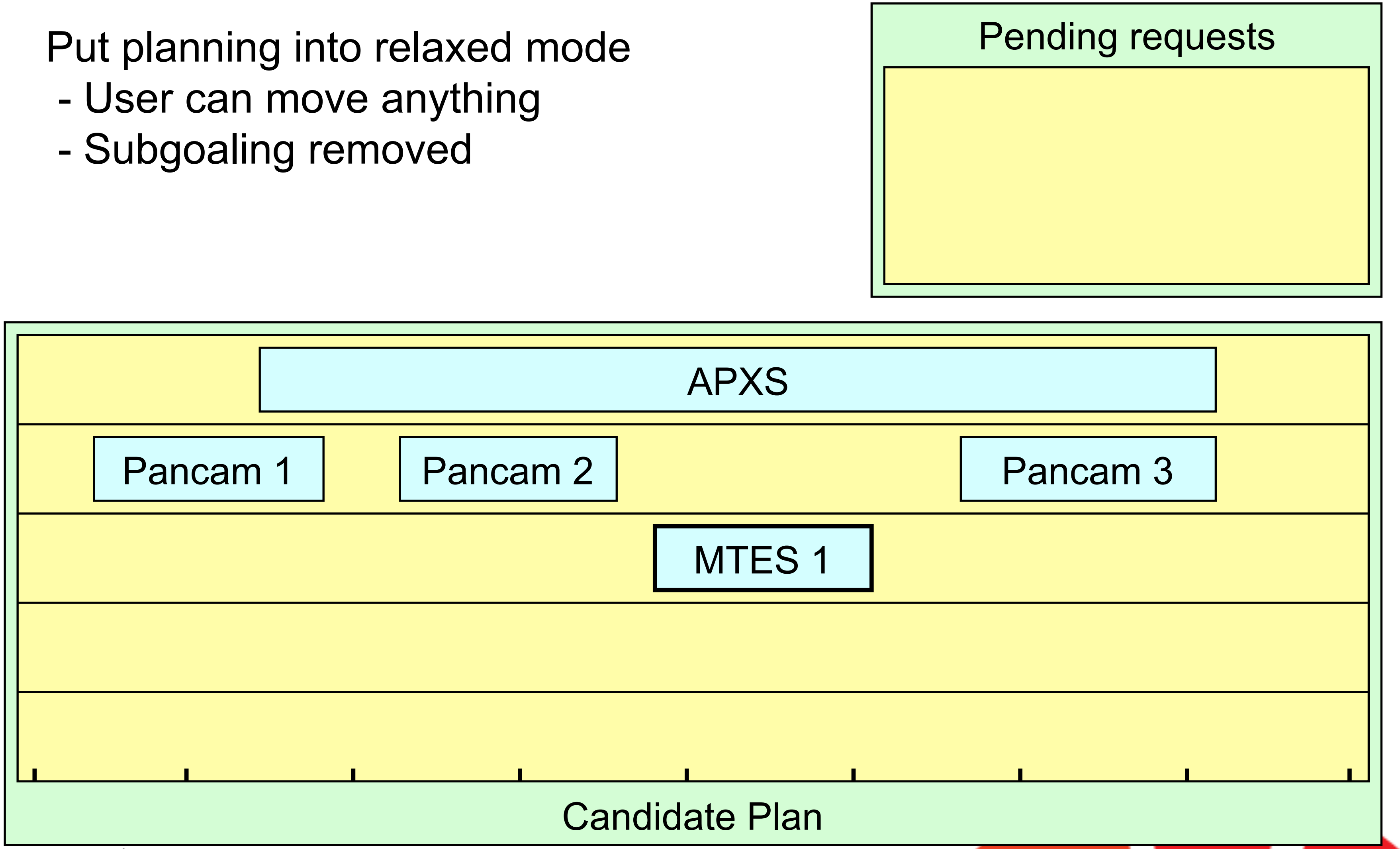
Example 5 - “Relaxed mode”

Want to try a different plan
-don’t like this one



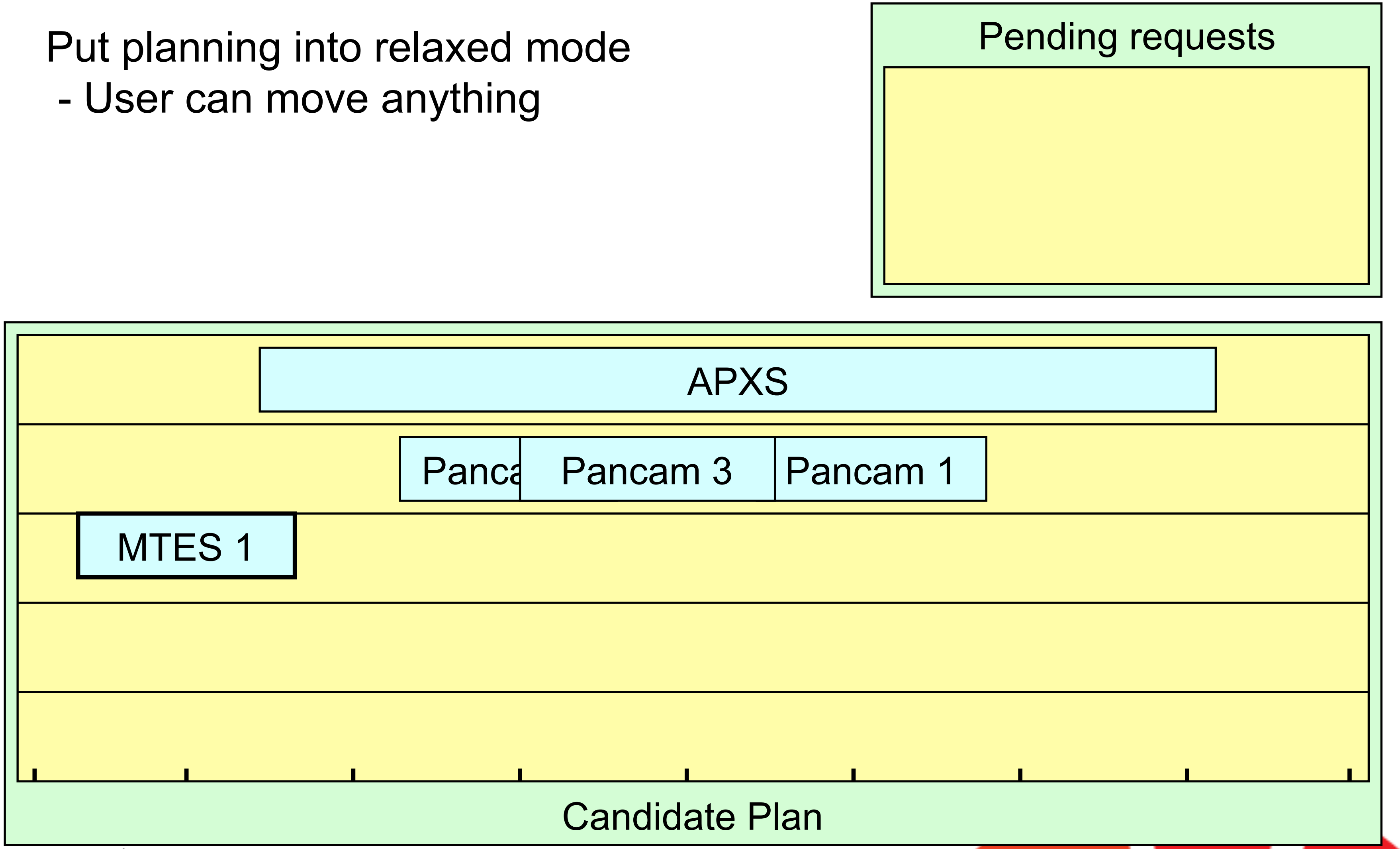
Example 5 - “Relaxed mode”

Put planning into relaxed mode
- User can move anything
- Subgoaling removed



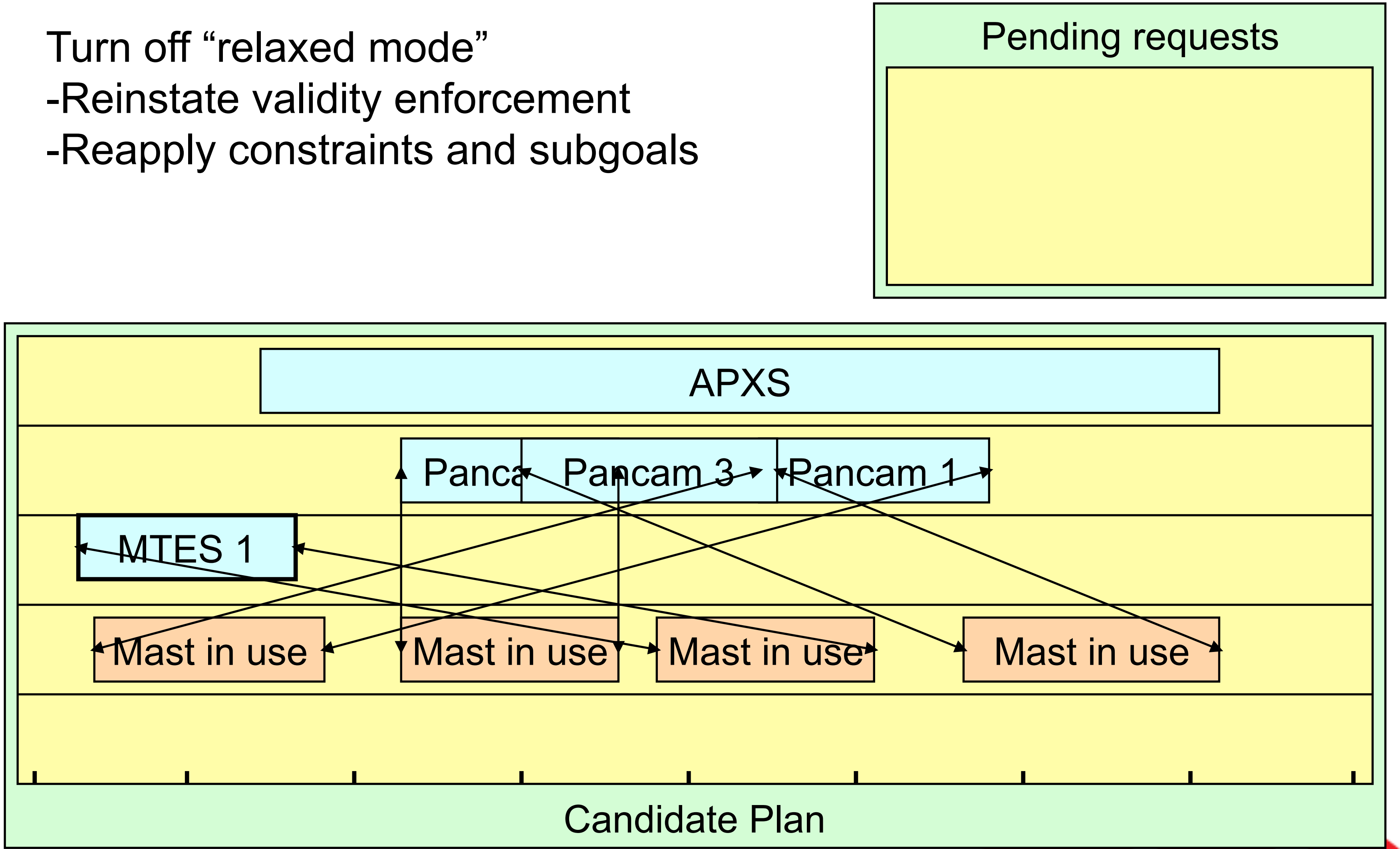
Example 5 - “Relaxed mode”

Put planning into relaxed mode
- User can move anything



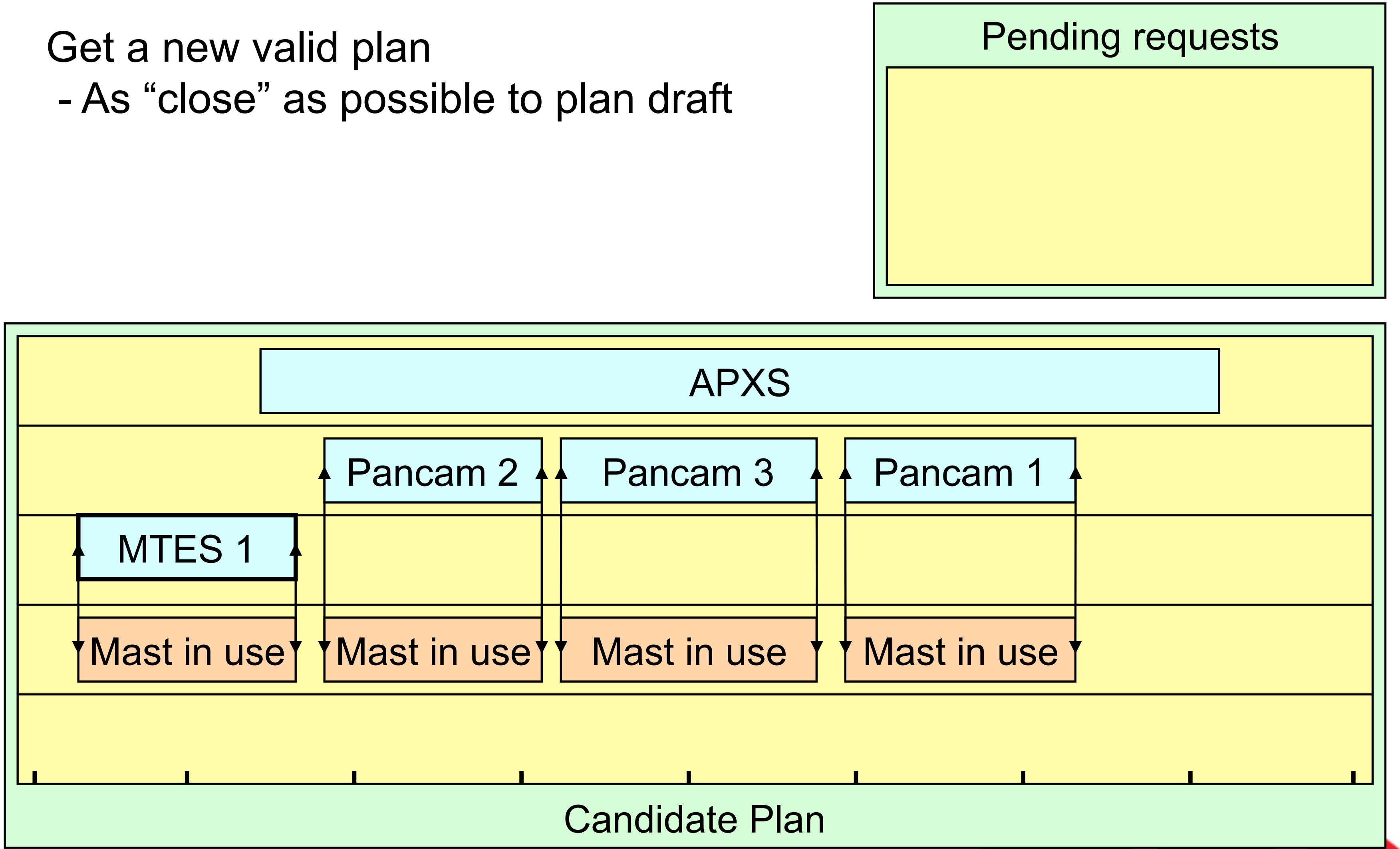
Example 5 - “Relaxed mode”

Turn off “relaxed mode”
-Reinstate validity enforcement
-Reapply constraints and subgoals



Example 5 - “Relaxed mode”

Get a new valid plan
- As “close” as possible to plan draft



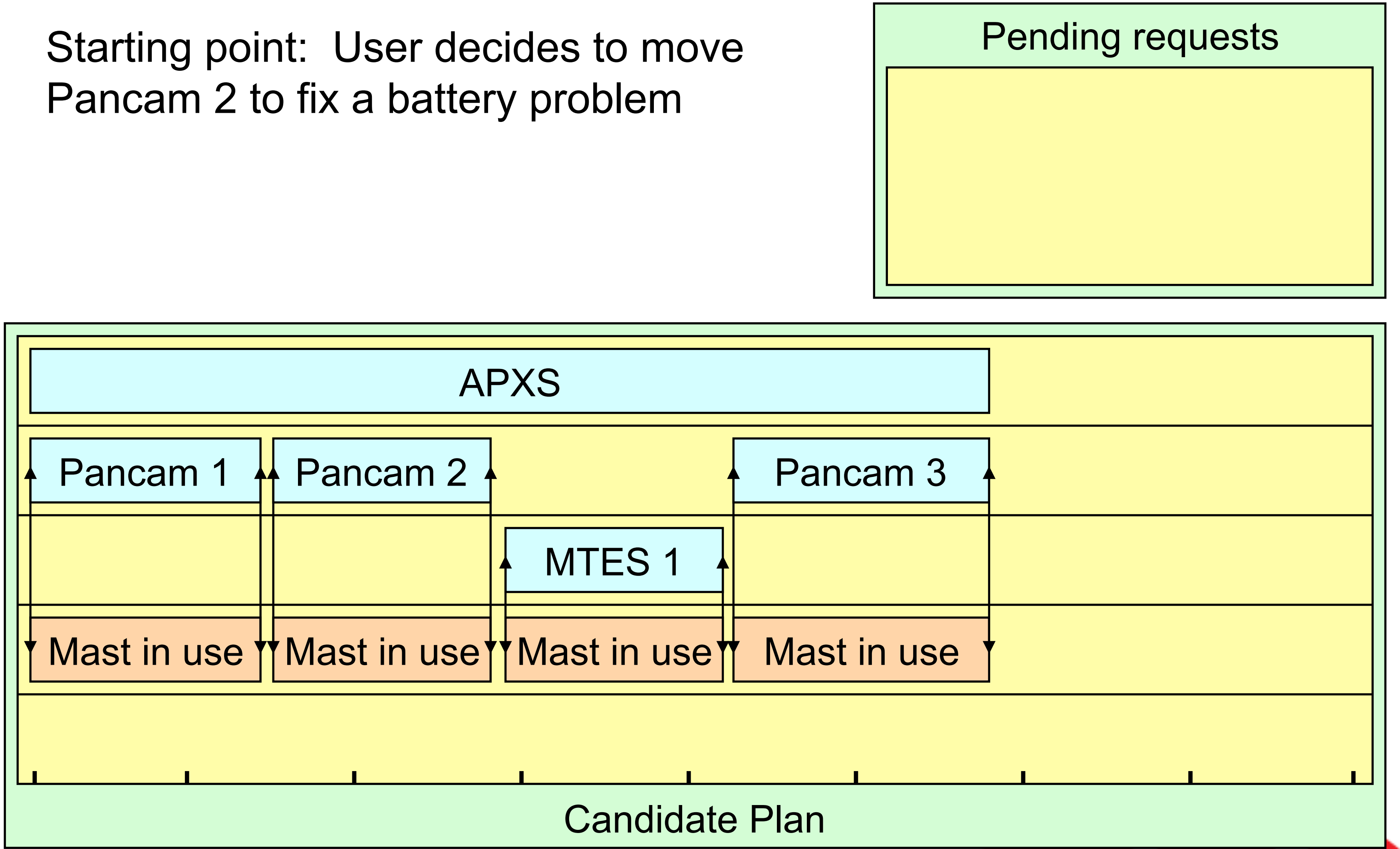
Time Handling

- Interesting issue
  - Underlying schedule is flexible (a simple temporal network)
  - Interface shows user a single instantiation
  - How to select that instantiation?
  - How to update it as modifications are made?

- Candidate solution: Earliest start time
  - Advantages: Known to be a valid solution
  - Disadvantages: Not very intuitive or user-friendly

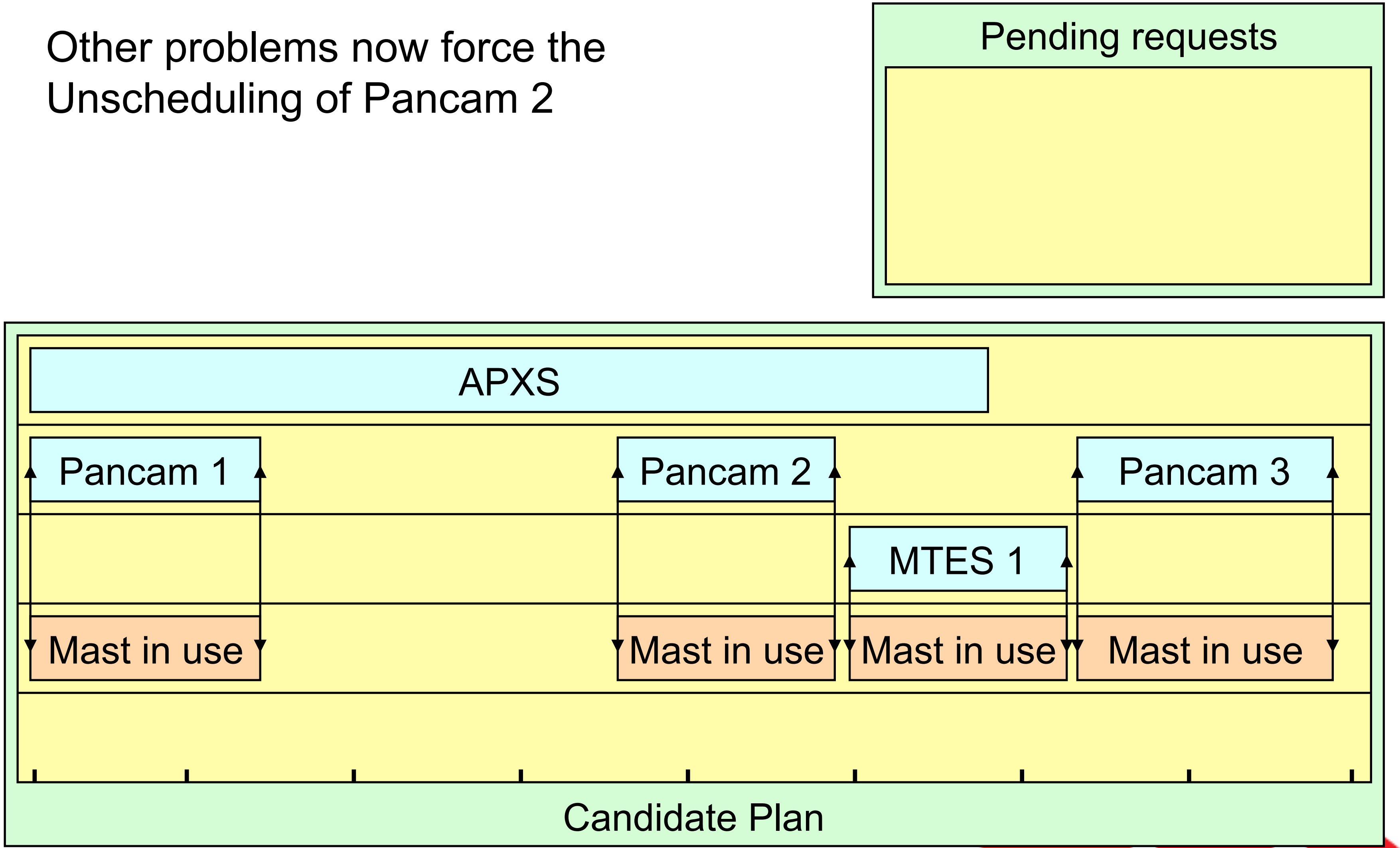
Using earliest start time

Starting point: User decides to move Pancam 2 to fix a battery problem



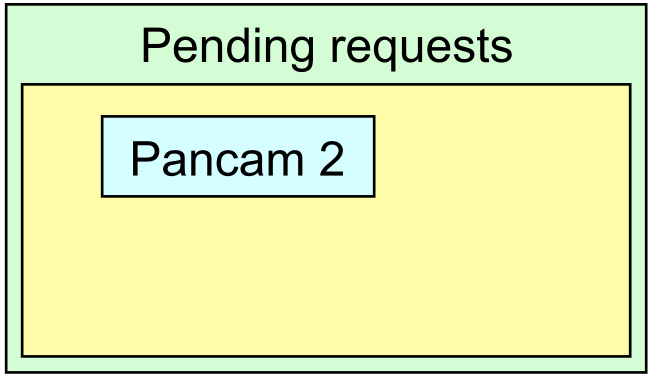
Using earliest start time

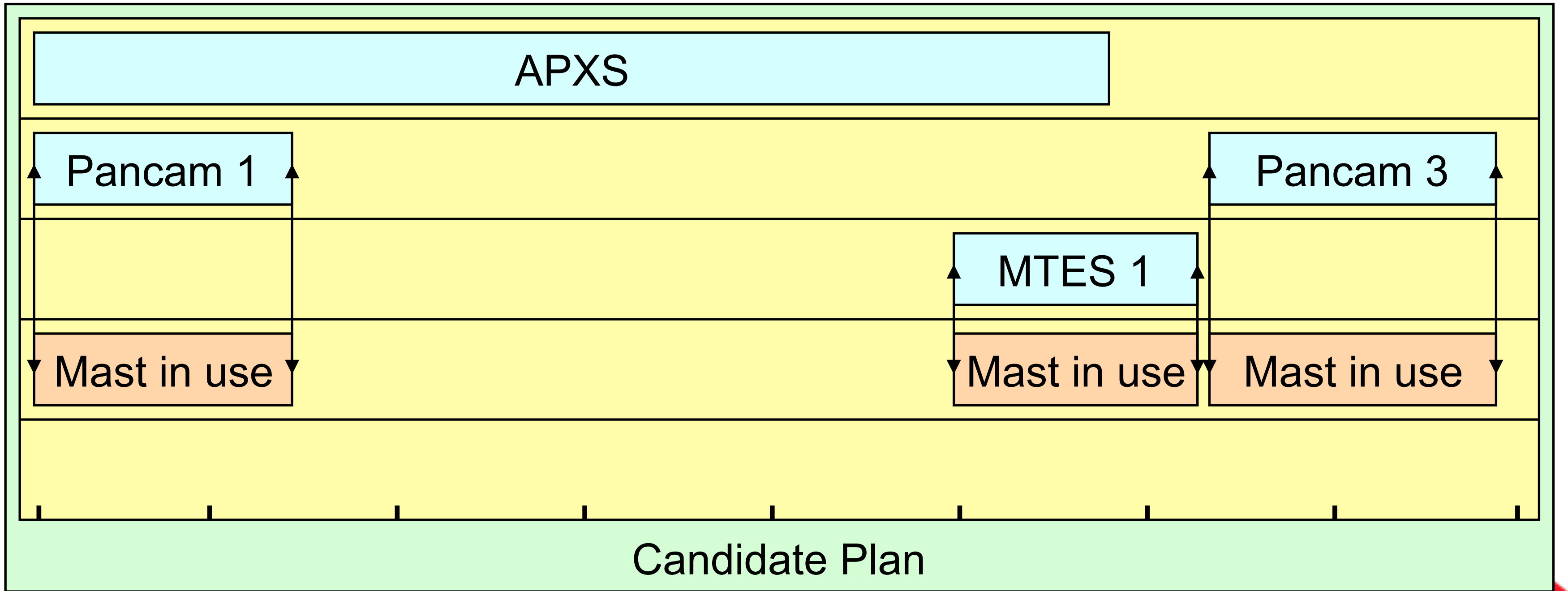
Other problems now force the Unscheduling of Pancam 2



Using earliest start time

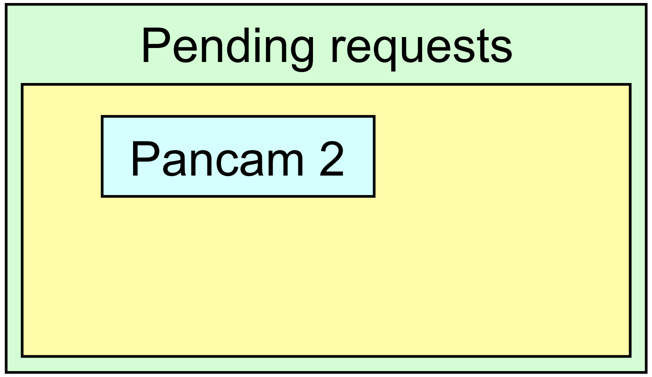
Other problems now force the
Unscheduling of Pancam 2

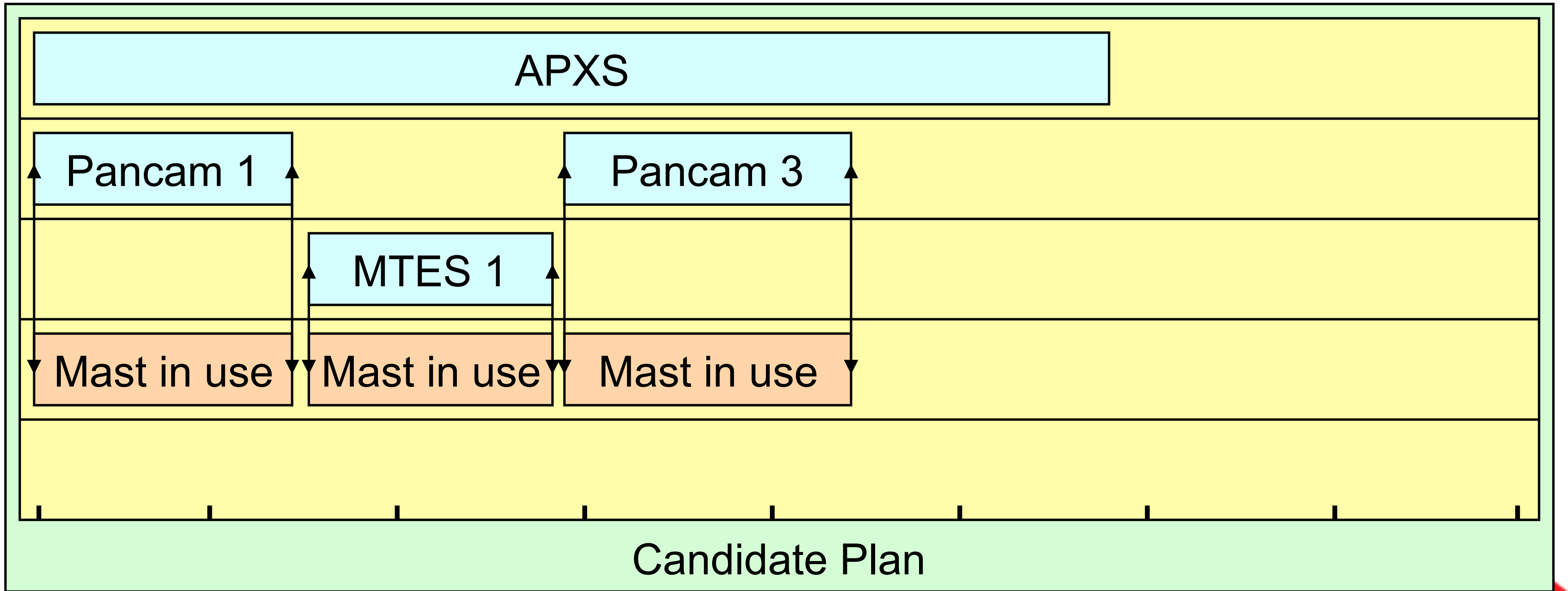




Using earliest start time

Earliest start time instantiation moves everything to the left



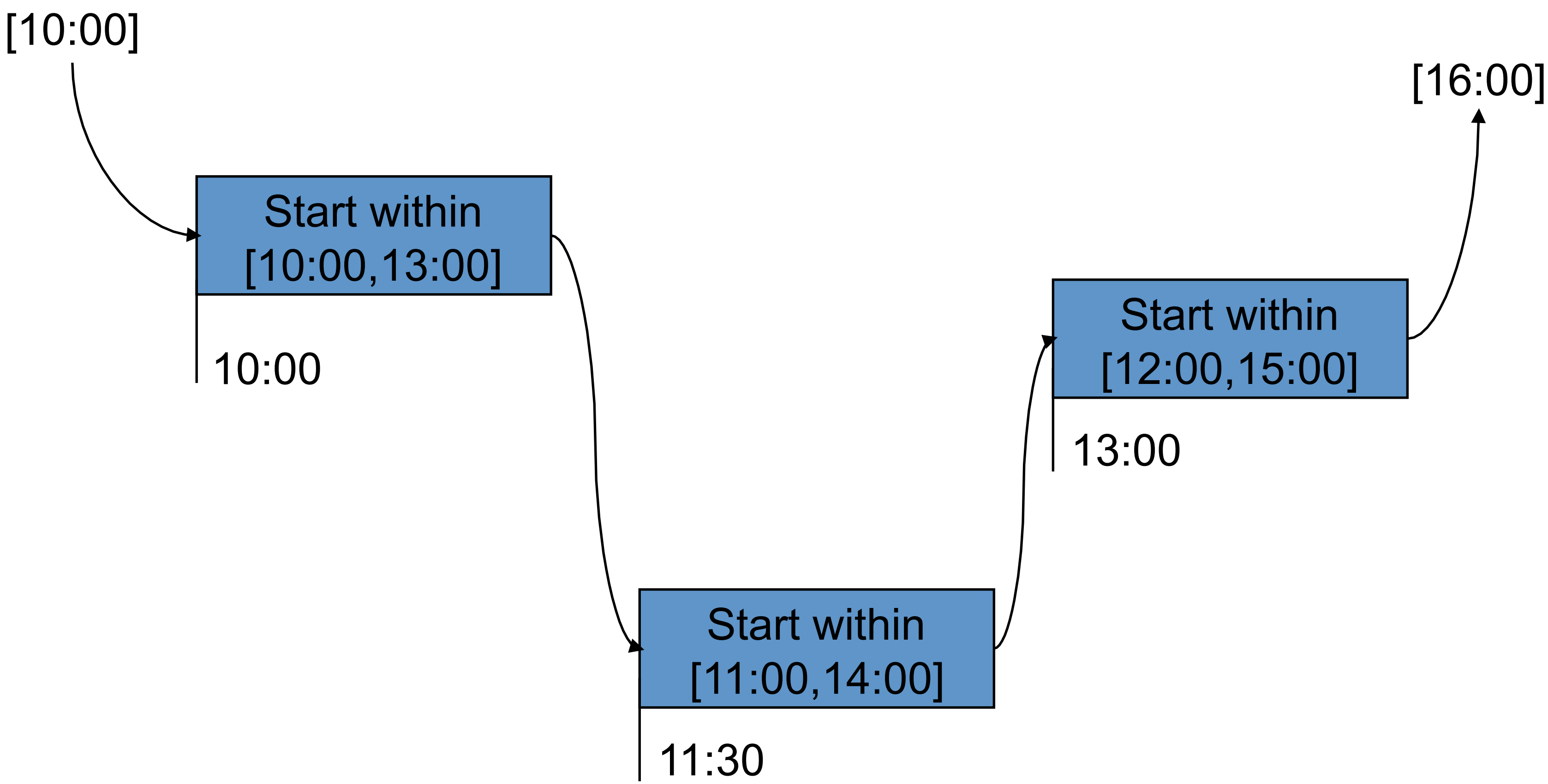


Better time handling required

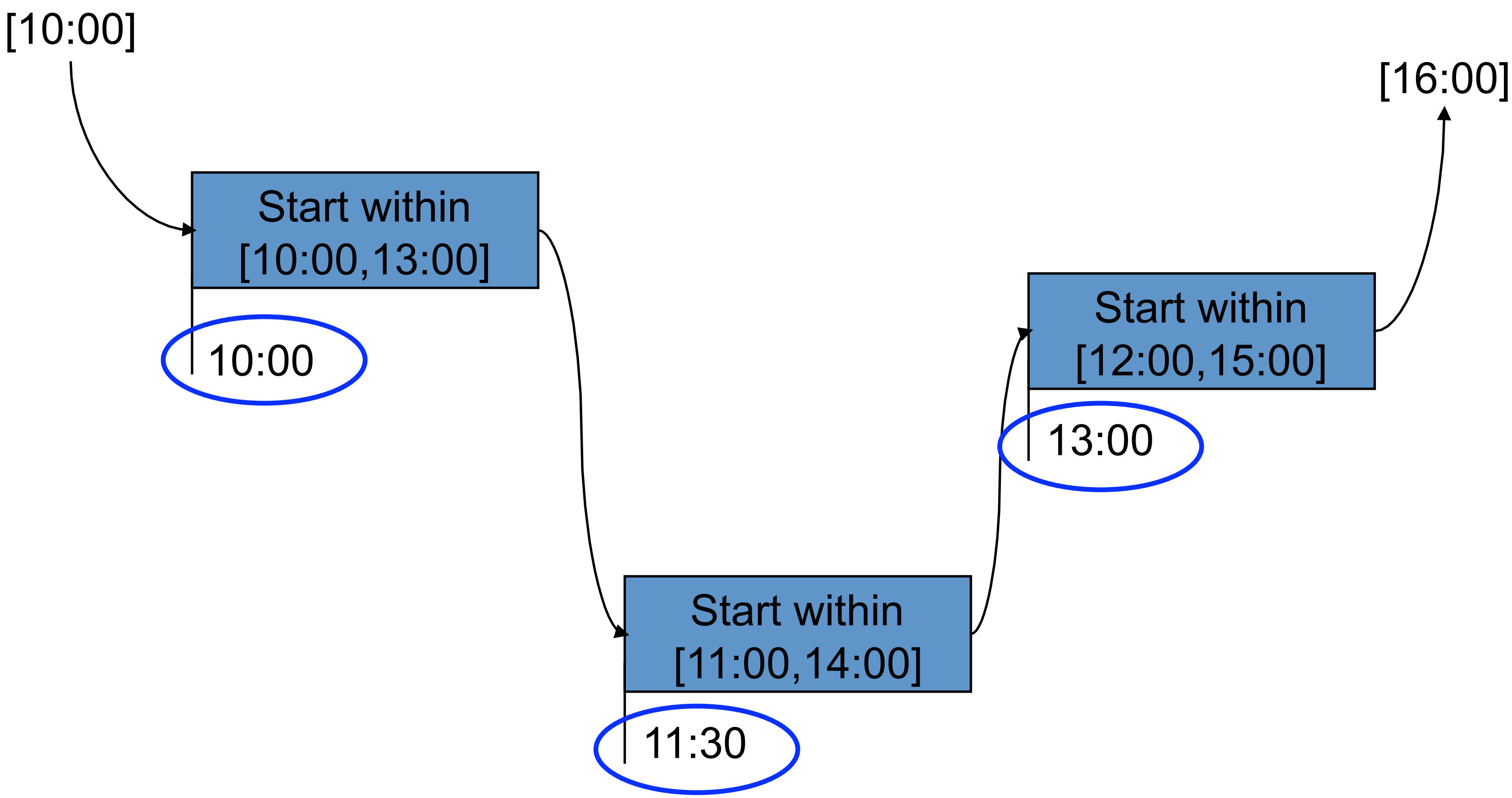
- Earliest start time is unintuitive for user
  - Small changes in plan lead to large changes in activity placement
- User move semantics when using earliest start time
  - Interpreted as constraints, which must be overridden if needed
  - User has no access to these constraints

- New approach:
  - Leave activities where they are whenever possible
  - Otherwise, change as little as possible
  - View placement and moves by user as preferences on placement
  - Absolute placements done by editing constraints
- Users were solidly in favor of this approach

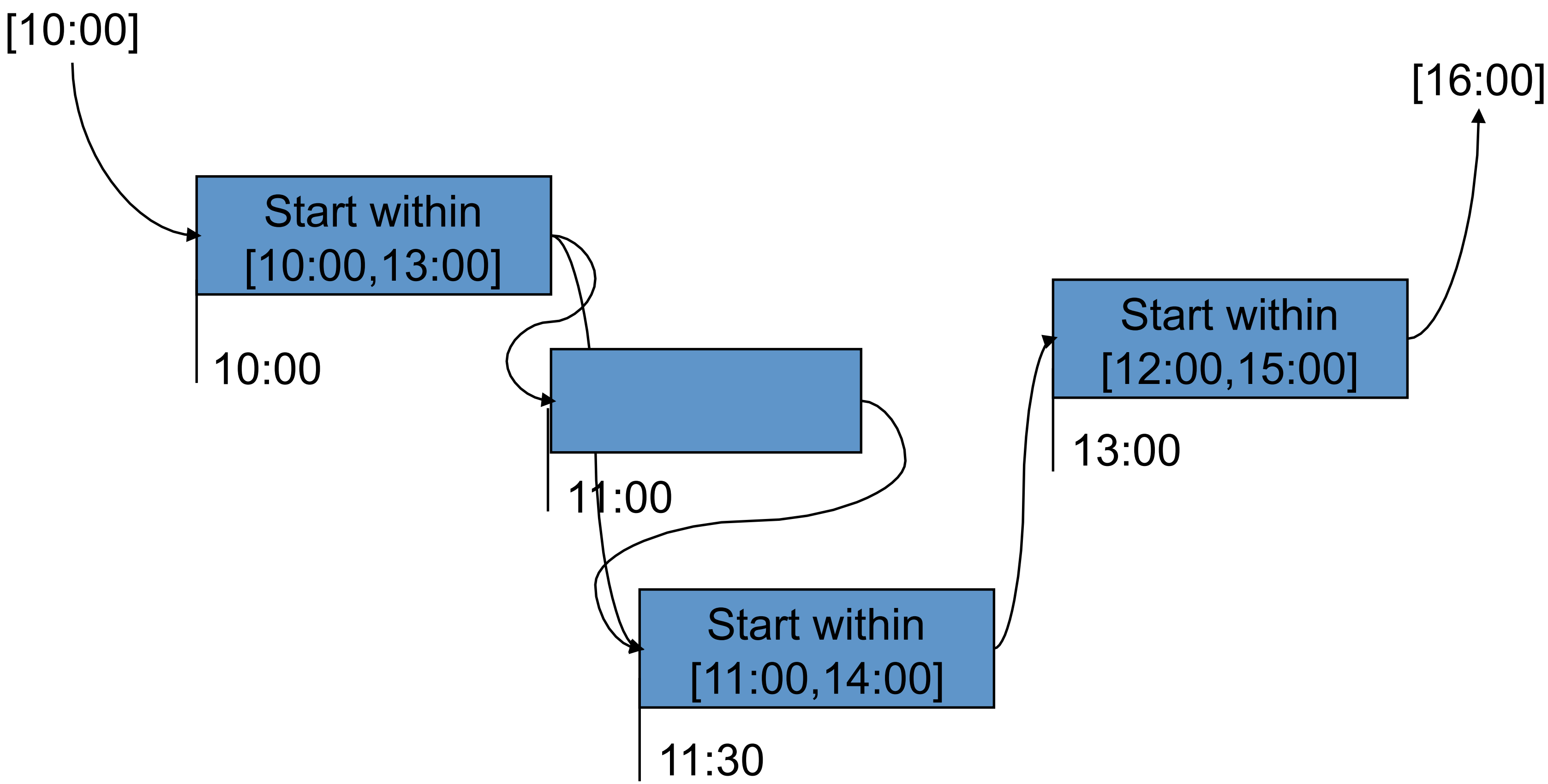
Temporal activity network



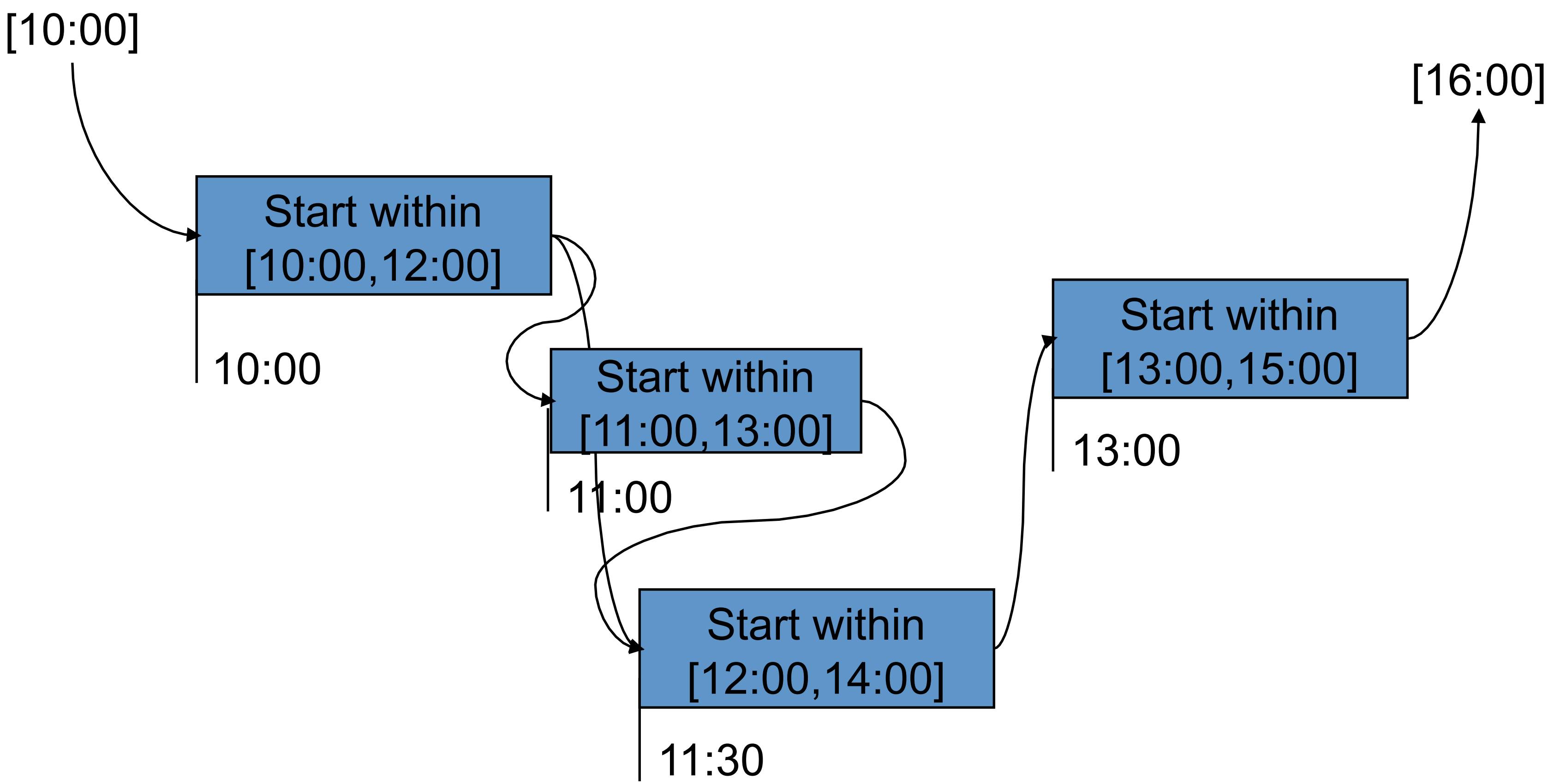
Current instantation



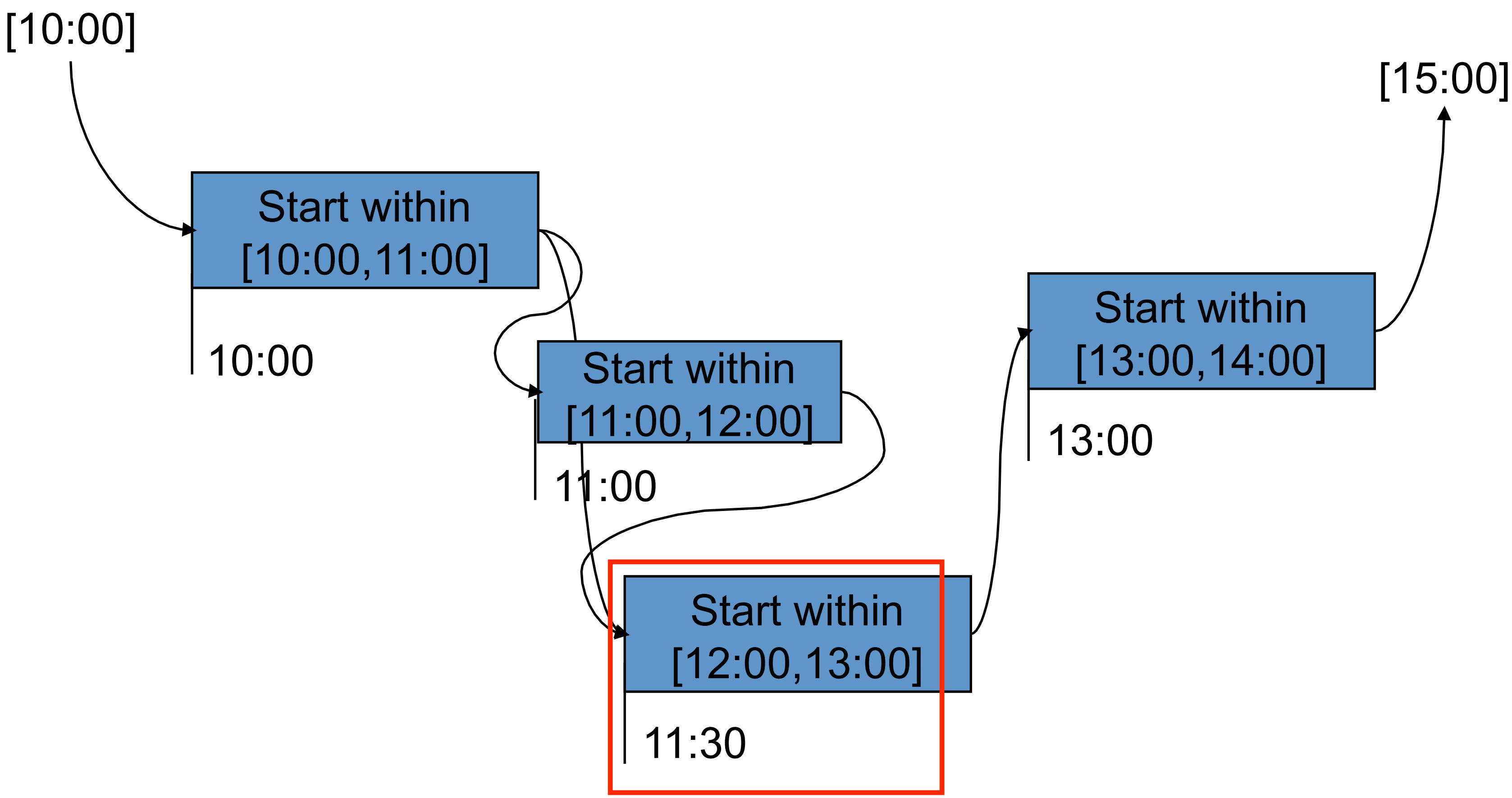
Add new activity and constraint



Propagate temporal network

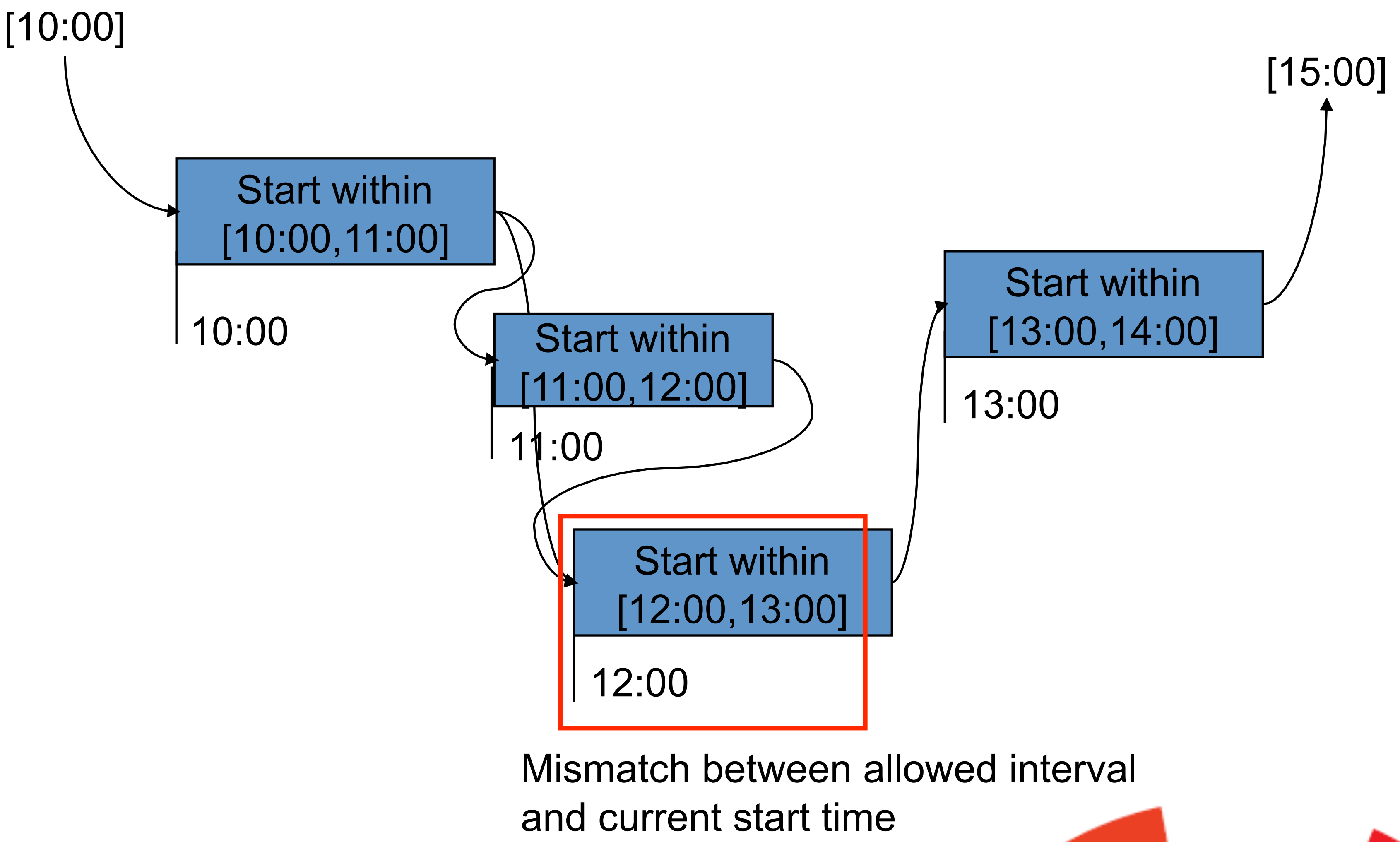


Propagate temporal network



Mismatch between allowed interval and current start time

Adjust start times out of bounds



Minimal perturbation algorithm

- Step 1: Remove all the current position constraints and re-propagate.
- Step 2: For each timepoint x with preferred position t do:

  if t is within the STN bounds for x

  then add a position constraint setting x to t

  else if t < the lower bound (lb) for x

  then add a position constraint setting x to lb

  else if t > the upper bound (ub) for x

  then add a position constraint setting x to ub

  propagate the effect of the new constraint
- Step 3: Update the preferred positions to the current ones.

So, how has this worked?

RESULTS

How did MAPGEN do?

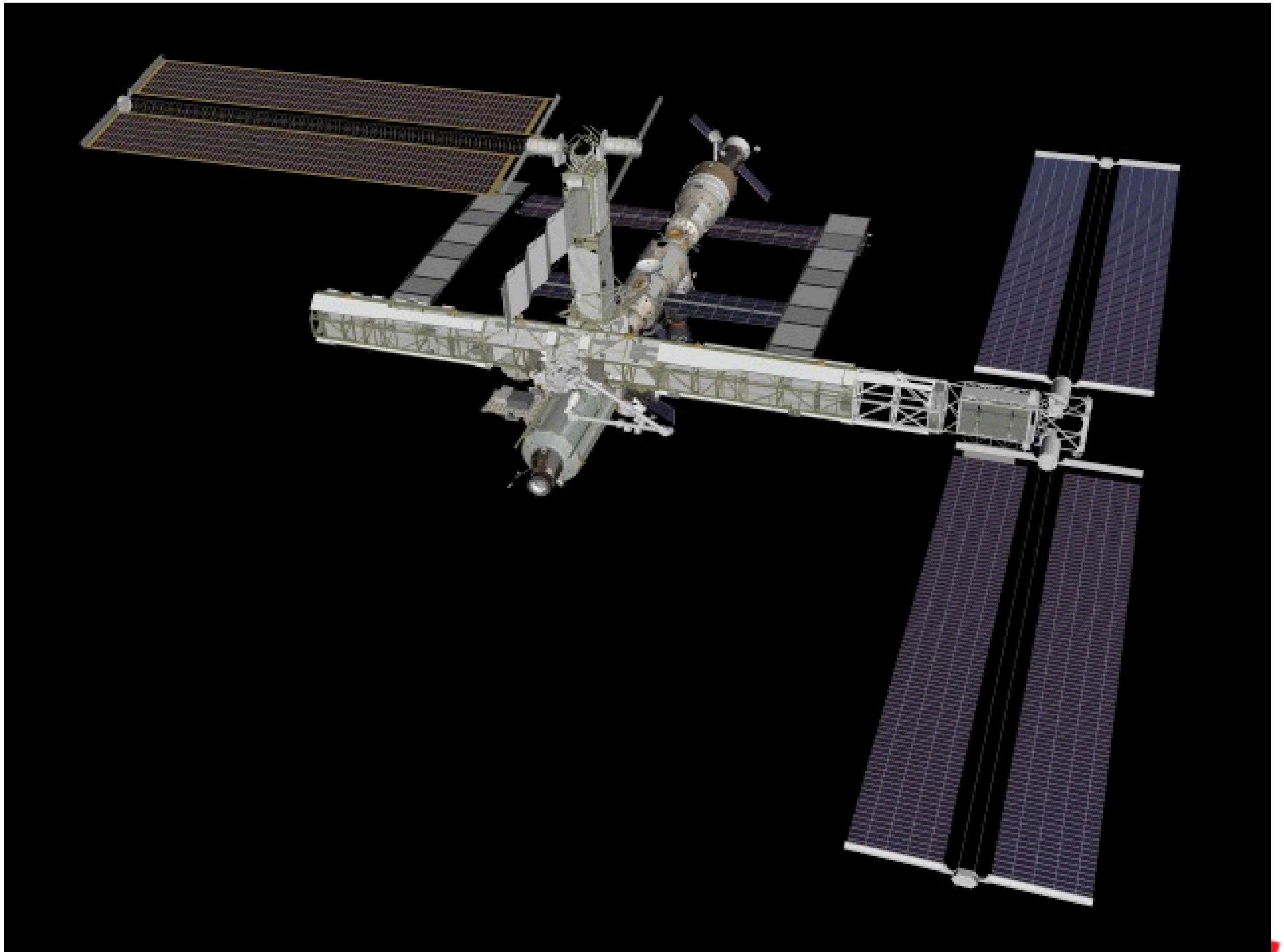
- Used throughout mission – still in use!
- Had a key role in early mission
- Evaluation:
  - Impossible to evaluate analytically
    - But, planning time often fraction of planned 2 hours
  - Subjective evaluations by mission folks
    - Increased science by 15-40% during initial mission

Many uses for EUROPA

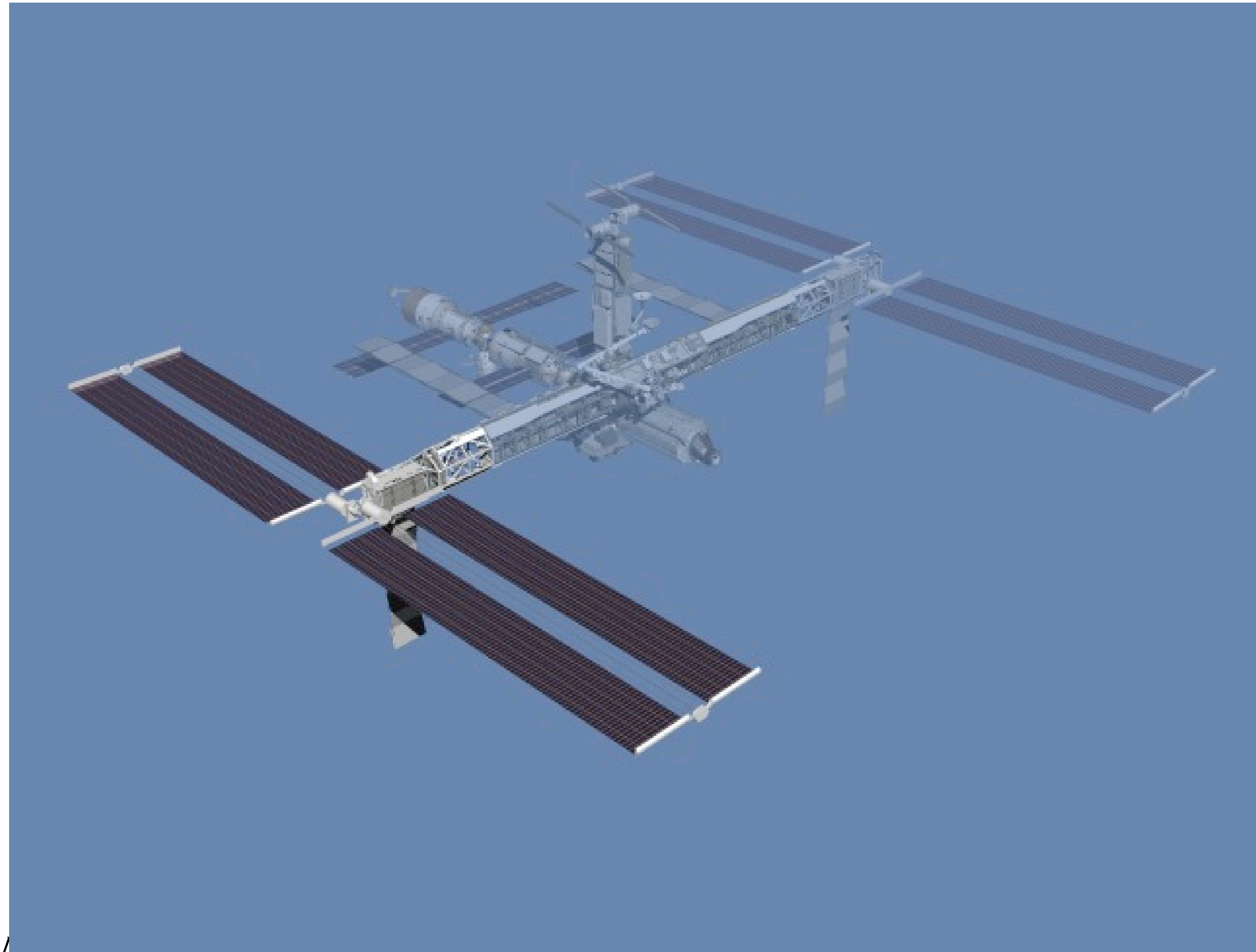
- ISS solar panels
  - Solar Array Constraint Engine
- Crew scheduling
  - Work under way at NASA Ames
- Robotic control
  - Atacama Desert Experiment
  - Willow Garage Robotics Architecture
- And more

Station before 12A

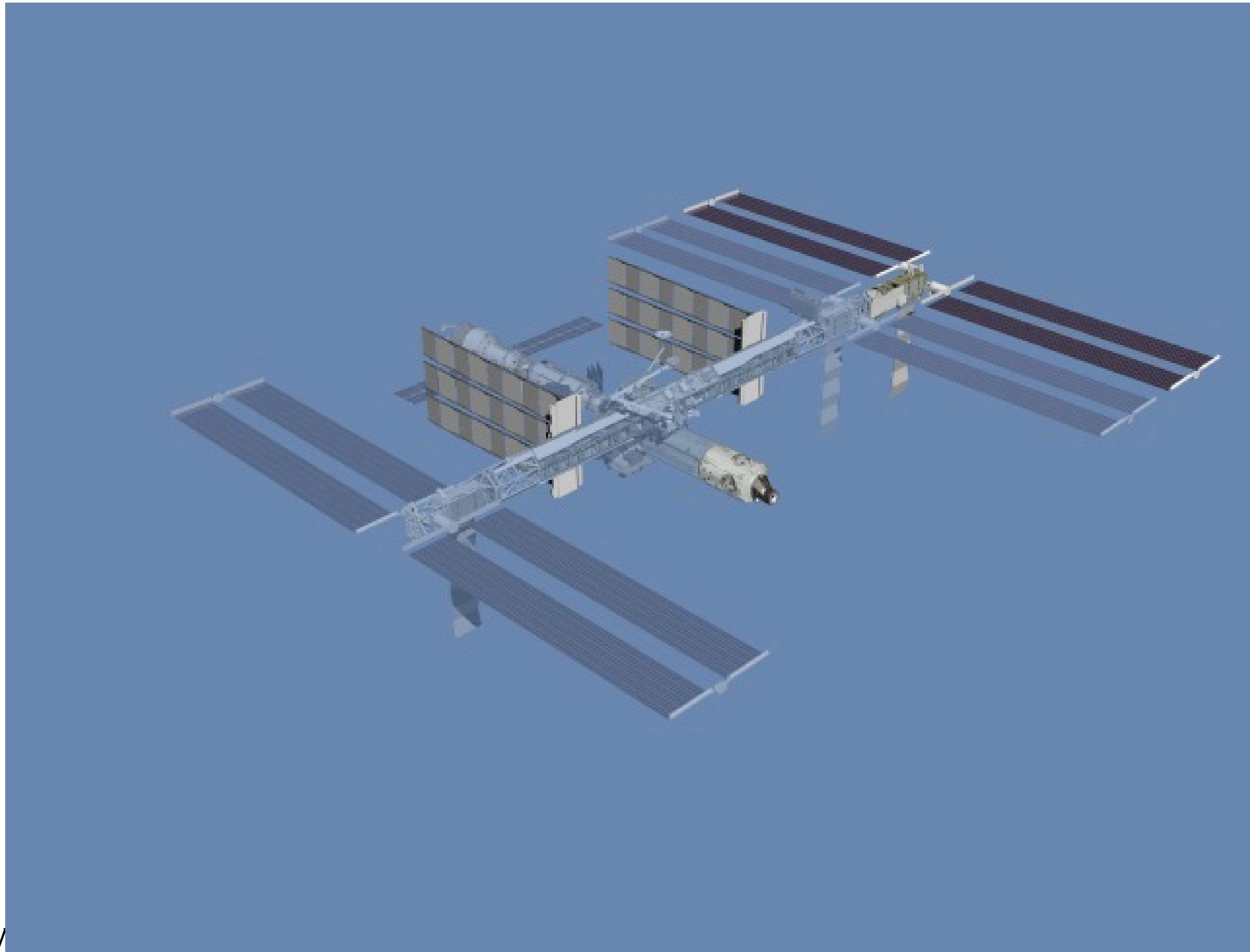
After 12A.1



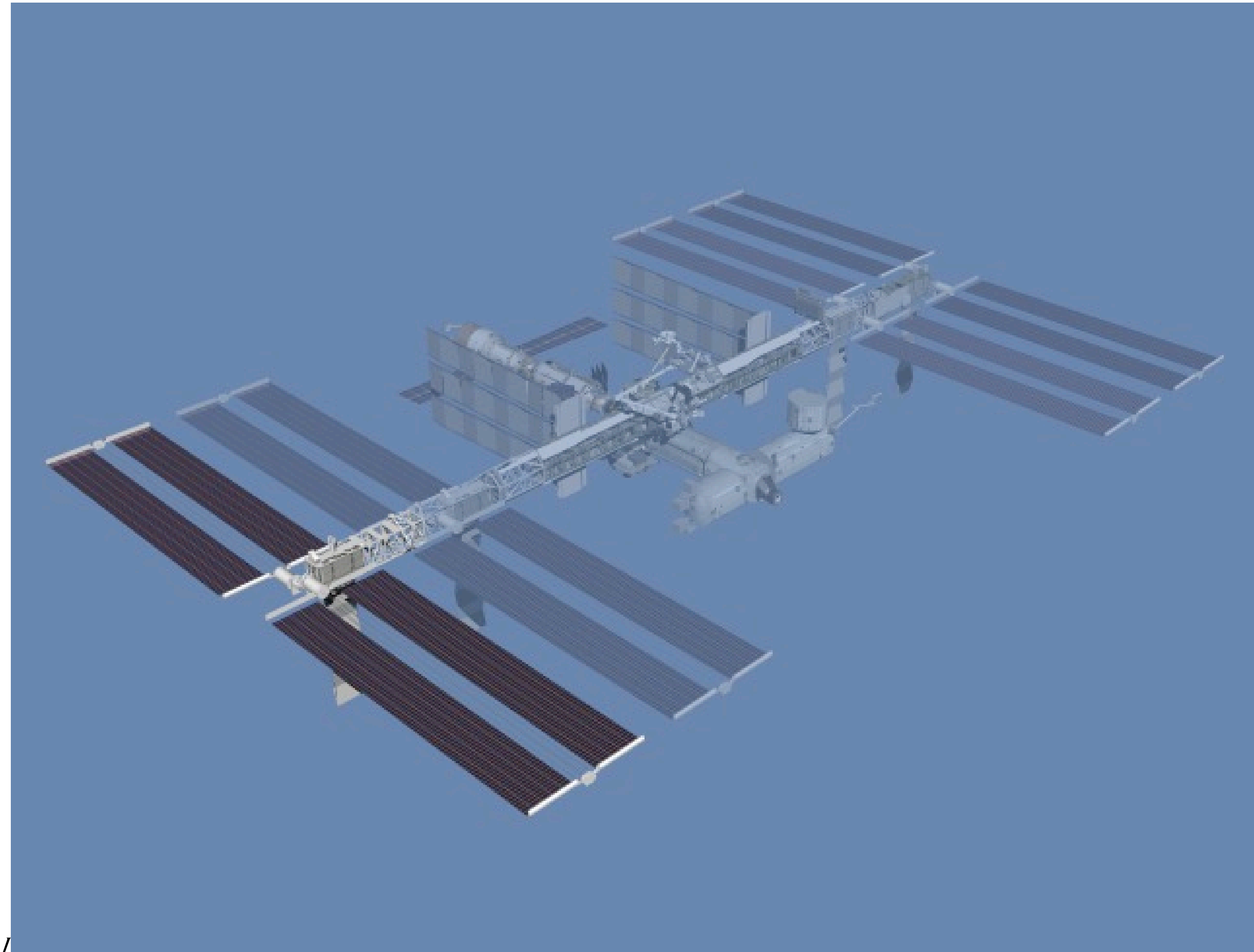
After 13A



After 10A - October 2007



After 15A - July 2008



Solar Array Operations

- Power Generation
  - Solar angles - alpha (daily) and beta (seasonal)
  - New solar panels articulate using alpha joint and beta joints
  - Power generation tightly related to solar array pointing
- Operational constraints
  - Operational modes
    - Station-keeping with different thruster selections
    - Visiting vehicles including docking and undocking
    - Water dumps, Debris-avoidance maneuvers, EVAs and more
  - Load limits
    - From thruster plumes
  - Shadowing limits
    - Structural longeron thermal loads
  - Contamination limits
    - Water dumps, thrusters, etc.

P4-2A

Alpha / Beta	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170
0	7.7	4.2	1.9	0.7	0.3	1.6	3.9	7.6	12	16	18	18	17	15	11	9.3	8.7	8.2
15	239	126	52	23	5.1	1.5	3.7	7	12	15	41	99	243	389	477	446	413	339
30	370	195	116	54	6.4	0.9	2.7	4.8	38	201	362	689	891	975	914	834	702	538
45	347	198	106	17	0	0.2	1.1	6.6	41	87	131	213	297	369	430	466	472	445
60	55	32	8.3	0	0	0	0.2	1.4	4.2	11	20	42	74	106	132	144	137	103
75	3.4	1.1	0.1	0	0	0	0	0.1	0.2	0.8	1.7	3.4	6	8.8	11	11	9.1	5.9
90	0.1	0	0	0	0	0	0	0	0	0	0.1	0.1	0.2	0.3	0.4	0.4	0.3	0.2
105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0
135	0	0	0.1	0.2	0.3	0.3	0.3	0.3	0.2	0.2	0.1	0	0	0	0	0	0	0
150	0	0.1	0.2	0.3	0.5	0.5	0.6	0.7	0.6	0.5	0.4	0.3	0.1	0	0	0	0	0
165	0	0.1	0.2	0.4	0.6	0.9	1.2	1.3	1.2	1.1	0.8	0.5	0.3	0.1	0	0	0	0
180	0	0.1	0.2	0.5	0.8	1.2	1.5	1.7	1.6	1.5	1.2	0.8	0.4	0.2	0.1	0	0	0
195	0	0.1	0.3	0.6	1	1.4	1.6	1.7	1.7	1.7	1.4	1	0.6	0.3	0.1	0	0	0
210	0	0.1	0.3	0.6	1	1.4	1.6	1.7	1.7	1.6	1.4	0.9	0.5	0.3	0.1	0	0	0
225	0	0.1	0.4	0.6	0.8	1.1	1.3	1.3	1.3	1.1	0.9	0.6	0.4	0.1	0	0	0	0
240	0	0.1	0.3	0.5	0.6	0.7	0.7	0.6	0.6	0.5	0.4	0.3	0.1	0	0	0	0	0
255	0.1	0.2	0.4	0.4	0.4	0.4	0.3	0.2	0.2	0.2	0.1	0	0	0	0	0	0	0
270	0.9	1.5	1.7	1.7	1.7	1.2	0.7	0.3	0.1	0	0	0	0	0	0	0	0.2	0.5
285	20	21	20	18	13	8.5	4.4	2.3	1.1	0.4	0.1	0	0	0	1.5	4.4	9.9	16
300	58	81	91	88	73	52	33	19	10	6.6	2.8	1.2	3.7	9.4	16	26	34	40
315	86	113	139	143	152	155	137	103	57	30	4.5	6.8	12	20	32	41	45	54
330	82	128	147	154	172	131	77	30	10	8.6	11	13	13	16	24	32	38	42
345	11	8.1	4.7	4.1	2.4	1.2	3.1	6	10	14	16	16	15	13	11	12	12	12

Alpha / Beta	180	190	200	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350
0	7	3.7	1.6	0.6	0.3	1.6	3.8	7.5	12	16	18	18	17	15	11	8.7	8.7	8.7
15	219	114	45	20	4.6	1.4	3.6	6.8	12	15	34	86	220	372	476	459	436	363
30	352	184	108	50	8.6	0.8	2.6	4.7	33	185	351	695	920	>999	954	870	732	562
45	336	193	110	21	0	0.2	1	6.9	41	94	140	226	313	388	451	490	497	467
60	57	34	9.5	0	0	0	0.2	1.4	4.6	12	22	45	79	113	140	150	141	102
75	3.7	1.3	0.2	0	0	0	0	0.1	0.2	0.9	1.9	3.7	6.5	9.4	11	11	9	5.6
90	0.1	0	0	0	0	0	0	0	0	0	0.1	0.1	0.2	0.3	0.4	0.4	0.3	0.2
105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0	0
135	0	0	0.1	0.2	0.3	0.3	0.3	0.3	0.3	0.2	0.1	0.1	0	0	0	0	0	0
150	0	0.1	0.2	0.4	0.5	0.6	0.6	0.7	0.6	0.5	0.4	0.3	0.1	0	0	0	0	0
165	0	0.1	0.2	0.4	0.6	0.9	1.2	1.3	1.2	1.1	0.8	0.5	0.3	0.1	0	0	0	0
180	0	0.1	0.2	0.5	0.8	1.3	1.5	1.7	1.7	1.5	1.2	0.8	0.5	0.2	0.1	0	0	0
195	0	0.1	0.3	0.6	1	1.4	1.6	1.7	1.7	1.7	1.4	1	0.6	0.3	0.1	0	0	0
210	0	0.1	0.3	0.6	1	1.4	1.6	1.7	1.7	1.6	1.3	0.9	0.5	0.3	0.1	0	0	0
225	0	0.1	0.4	0.6	0.8	1.1	1.2	1.3	1.2	1.1	0.9	0.6	0.4	0.1	0	0	0	0
240	0	0.1	0.3	0.5	0.6	0.7	0.7	0.6	0.6	0.5	0.4	0.3	0.1	0	0	0	0	0
255	0.1	0.2	0.4	0.4	0.4	0.4	0.3	0.2	0.2	0.1	0.1	0	0	0	0	0	0	0
270	1	1.6	1.8	1.7	1.6	1.1	0.6	0.3	0.1	0	0	0	0	0	0	0	0.1	0.4
285	20	20	19	17	13	8.1	4	2.1	0.9	0.4	0.1	0	0	0	1.2	3.9	9.2	15
300	57	78	86	83	68	48	31	18	9.1	5.9	2.6	1.1	3.5	9.3	16	26	34	41
315	81	106	132	135	143	146	128	98	54	31	5.7	6.7	12	20	32	42	46	59
330	76	120	139	149	169	135	83	34	13	8.5	11	13	14	17	25	33	39	44
345	11	8.5	5.2	4.4	2.7	1.2	3	5.9	10	13	16	16	16	13	11	12	12	12

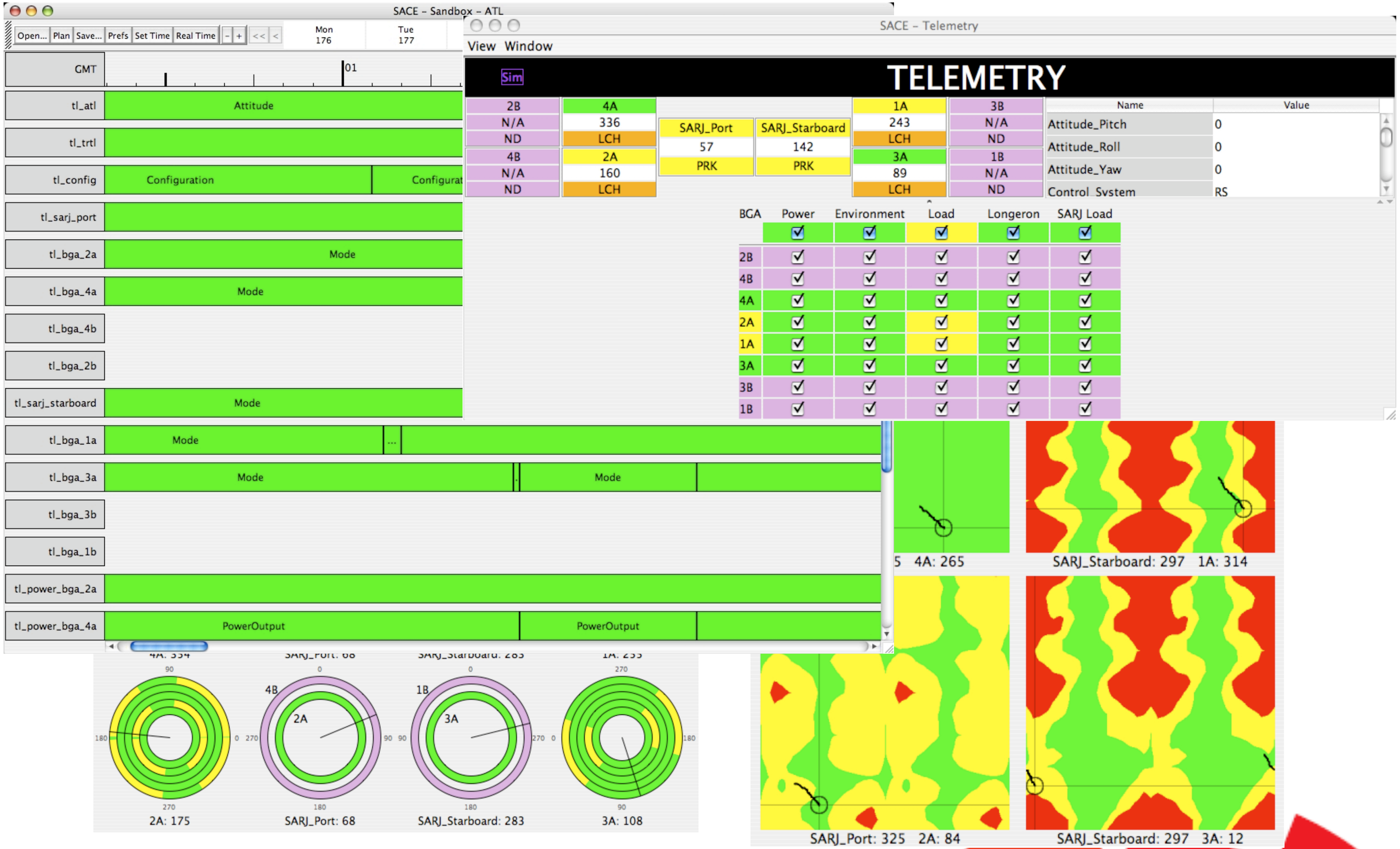
Solar Array Operations

- Lock/latch/park/track solar arrays
  - Complex constraints for when okay to track, when to lock/latch
    - Tracking depends on patch
    - Locking/latching depends on possible faults and drifts
- Contingency handling
  - Malfunctions in controllability
  - Malfunctions in position determination
- Preferences
  - Avoid locking SARJs and latching BGAs
  - Minimize slew times
- Solution trade-offs and choices
  - Find best solution that provides needed power
  - Get most power with no constraint violations
  - Find best solution that avoids locking/latching

Solar Array Operations

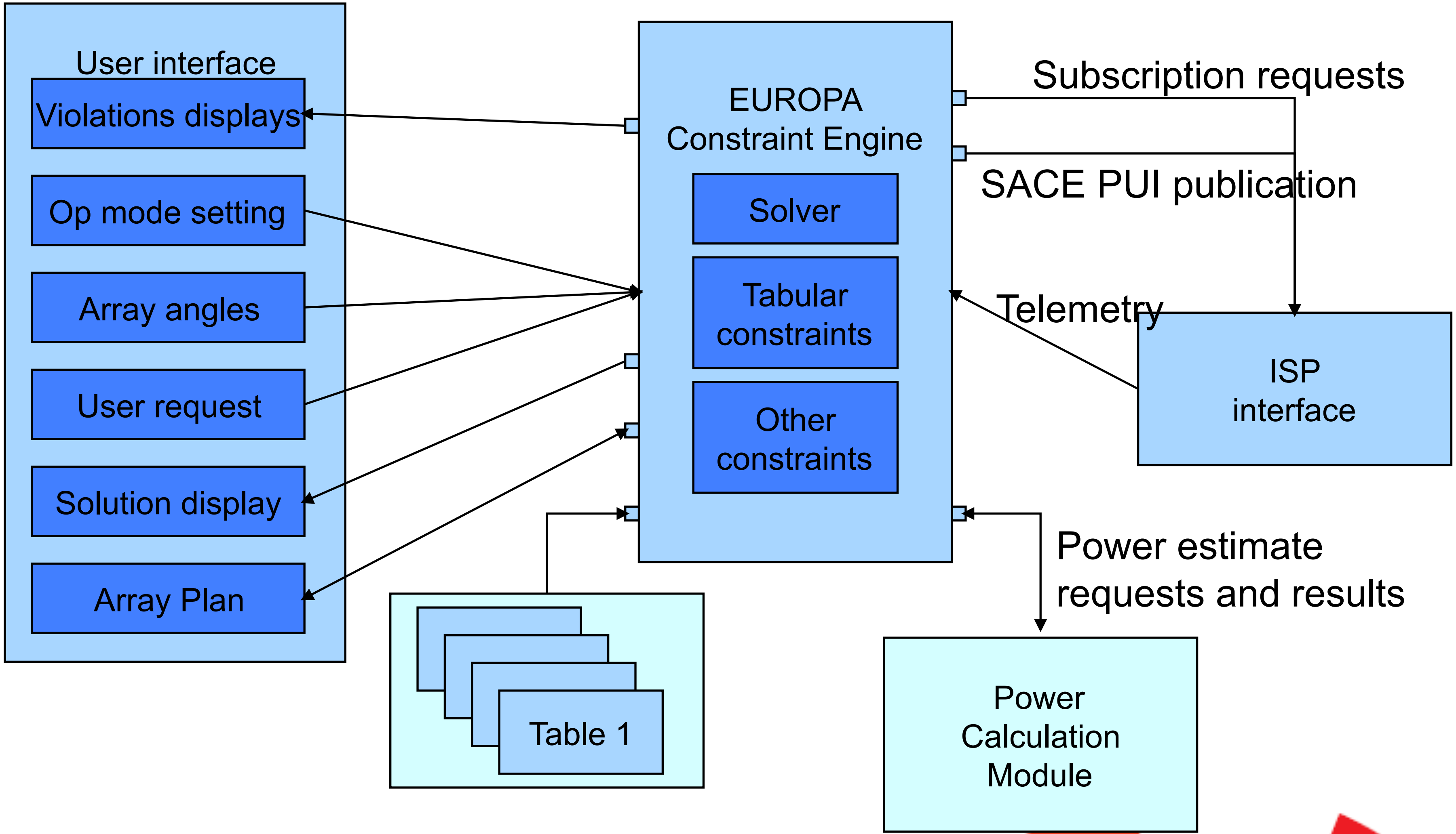
- Flight controllers (PHALCons) need support tools
  - Constraint interactions and prioritization
  - Balancing constraints and power generation needs
  - Responding safely to faults and unplanned situations
- Solar Array Constraint Engine
  - Handles multiple constraints sources
    - Hundreds (later thousands) of load/erosion/power tables
    - Longeron thermal constraints
    - Power generation calculations
  - Plugs into mission telemetry streams
    - Tracks state and determines applicable constraints and their status
  - Provides visual display interface to controller
    - Supports "what if" analysis to find safe configurations
    - Supports automated identification of optimal configurations
  - Provides solar array operations plan management
    - Automated generation of solar array plans
    - Solar array plan editing and "what if" interactions

SACE Interface



REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

SACE Architecture



What next? Where are we going?

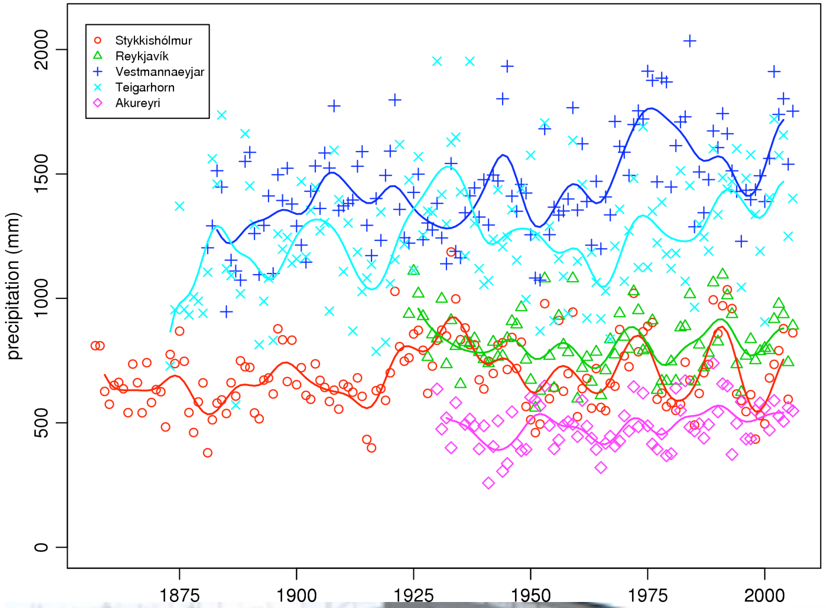
FUTURE WORK

What is ahead?

- Additional role in human space flight
- Upcoming MSL mission to Mars
- Role in robotics architecture
- Potential use in many applications
  - Controlling non-playing characters
  - Handling communications planning
  - Assist with power system operations
  - Manage automatic sensors

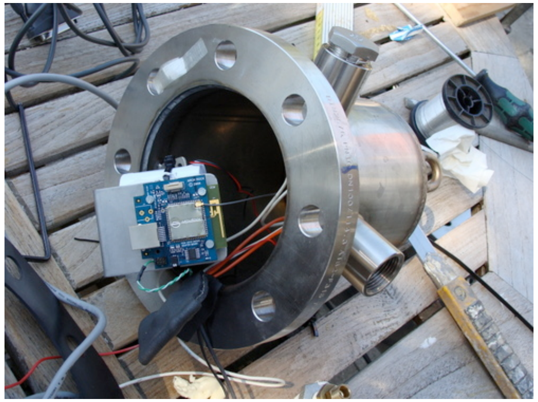
Non-playing Characters in Games

Power System Operations



REYKJAVÍK UNIVERSITY
HÁSKÓLINN Í REYKJAVÍK

Automated Sensors



But, EUROPA is not perfect

- Hard to deal with violations in plan
- Resources tricky, especially in search
- General heuristics not good yet
- Large and complex software

- Partial explanation:
  – Very expressive and general

Key challenges

- Packaging of “practical planning” tools
  - EUROPA is available as open source
    - But, much more work is needed
  - Very hard to use “out of the box”
- Applicability to “end users”
  - Development requires planning expertise
- User interface support
  - Explanations, reports, etc.

Key challenges

- Search control
  - Not just for finding complete plans
- Easier configurability
  - Specification of reasoning scope
  - Preferences and exceptions
- Modeling
  - Requires expertise and “insider information”
  - Needs to become realm of engineers

Continued development

- Heuristics
  - Some progress being made recently
- Resources
  - Fairly good handling in grounded cases
  - Difficult in partial commitment cases
- Simplification
  - Modularization and accessability
- Unification with other systems
  - Both within state variable planning and with classical planning approaches