Classical Planning: Limits

Instantaneous actions

No temporal constraints

No concurrent actions

No continuous quantities

Spacecraft Domain

Observation-1
- priority
- time window
- target
- instruments
- duration

Observation-2

Observation-3

Observation-4

…

Objective:
maximize science return

Spacecraft Domain

linked

Observation-1
- priority
- time window
- target
- instruments
- duration

Observation-2

Observation-3

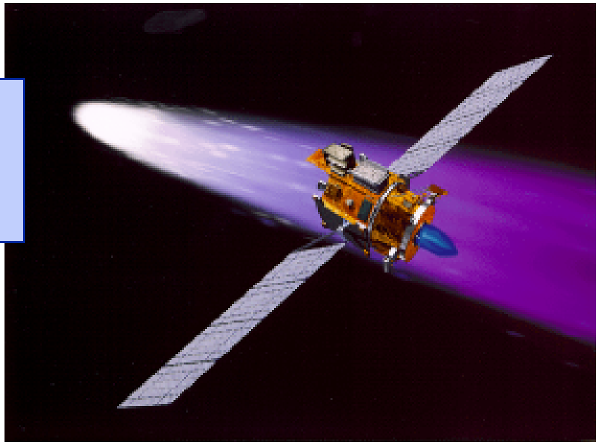
Observation-4

…

angle between targets
⇒ turn duration

calibration
- target1
- target2
- …

consumables:
- fuel
- power
- data storage
- cryogen



Objective:
maximize science return

Based on slides by Dave Smith, NASA Ames

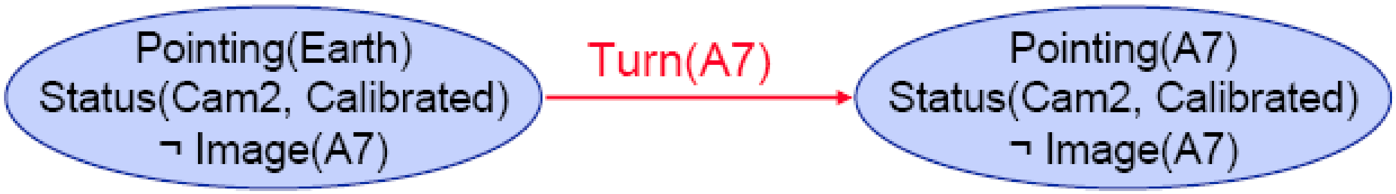
Extensions

- Time
- Resources
- Constraints
- Uncertainty
- Utility
- …

Model

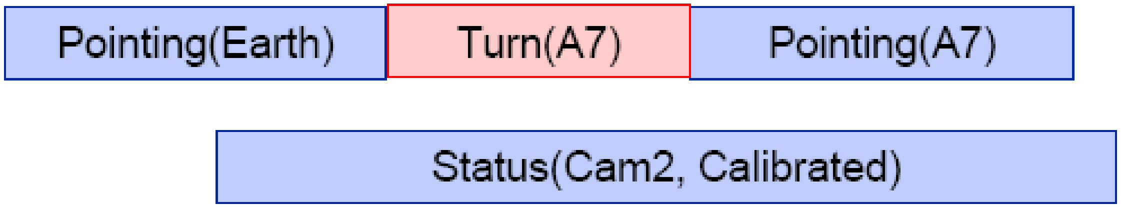
State-centric (Mc Carthy):

for each time describe propositions that are true



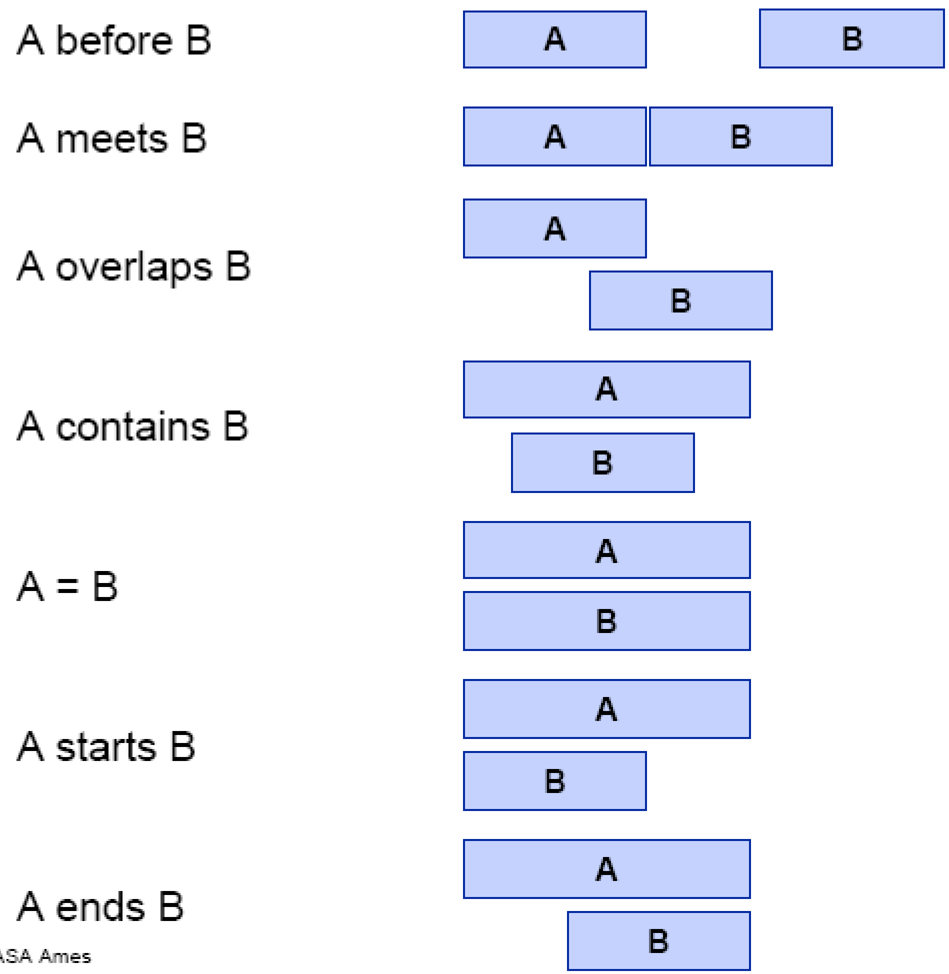
History-based (Hayes):

for each proposition describe times it is true



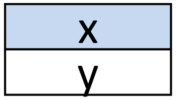
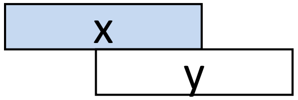
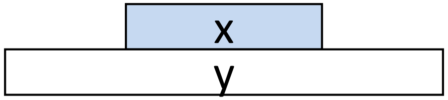
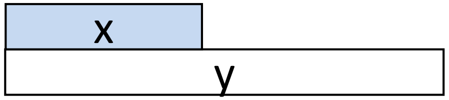
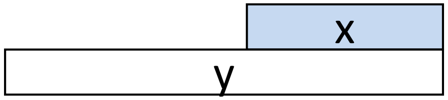
Based on slides by Dave Smith, NASA Ames

Temporal Interval Relations



Based on slides by Dave Smith, NASA Ames

Interval Algebra (aka Allen Algebra) [Allen 83]

Relation	Symbol	Inverse	Illustration
X before Y	b	bi	
X equal Y	=		
X meets Y	m	mi	
X overlaps Y	o	oi	
X during Y	d	di	
X starts Y	s	si	
X finishes Y	f	fi	

Interval Algebra: Qualitative TN

- Variables
  - An interval represent an event with some duration
- Constraints
  - Intervals *I*, *J* are related by a binary constraint
  - The constraint is a subset of the 13 basic relations

    $r = \{ b, m, o, s, d, f, bi, mi, oi, si, di, fi, = \}$
  - Example: $I \{r_1, r_2, \ldots, r_k\} J \Leftrightarrow (I\ r_1\ J) \vee (I\ r_2\ J) \vee \ldots \vee (I\ r_k\ J)$
  - Enumerate atomic relations between two variables

Interval Algebra Constraint Network

- Variables: temporal intervals *I* and *J*
- Domain: set of ordered pairs of real numbers
- Constraints are subsets of the 13 relations
  - How many distinct relations?
- A solution is an assignment of a pair of numbers to each variable such that no constraint is violated

Interval Algebra: Example

Story:

John was not in the room when I touched the switch to turn on the light but John was in the room later when the light was on.

CSP model:

Variables:

Switch – the time of touching the switch

Light – the light was on

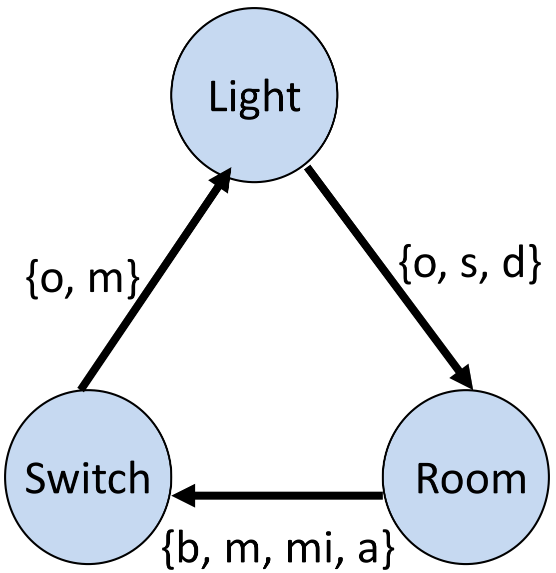
Room – the time that John was in the room

Constraints:

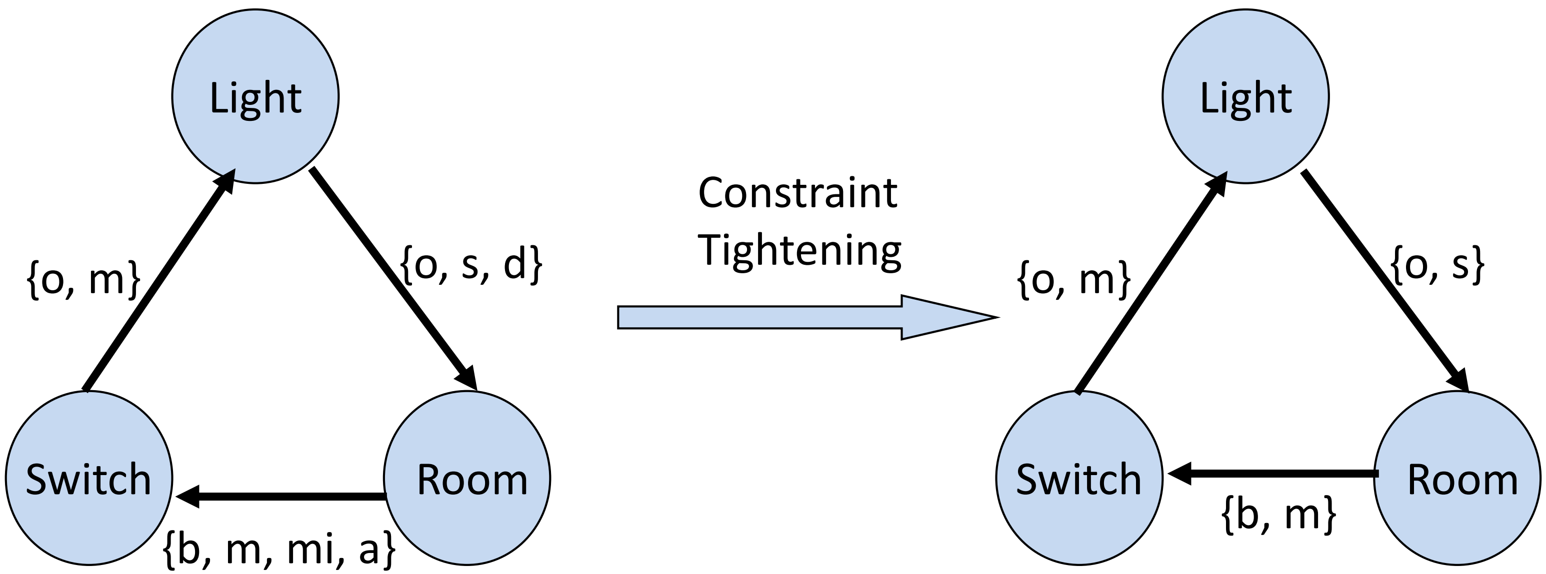
Switch overlaps or meets Light: *S* {*o, m*} *L*

Switch is before, meets, is met by or after Room: *S* {*b, m, mi, bi*} *R*

Light overlaps, starts or is during Room: *L* {*o, s, d*} *R*



The Task: Get the Minimal Network



A unique network equivalent to original network
All constraints are subsets of original constraints
Provides a more explicit representation
Useful in answering many types of queries

Temporal Operators

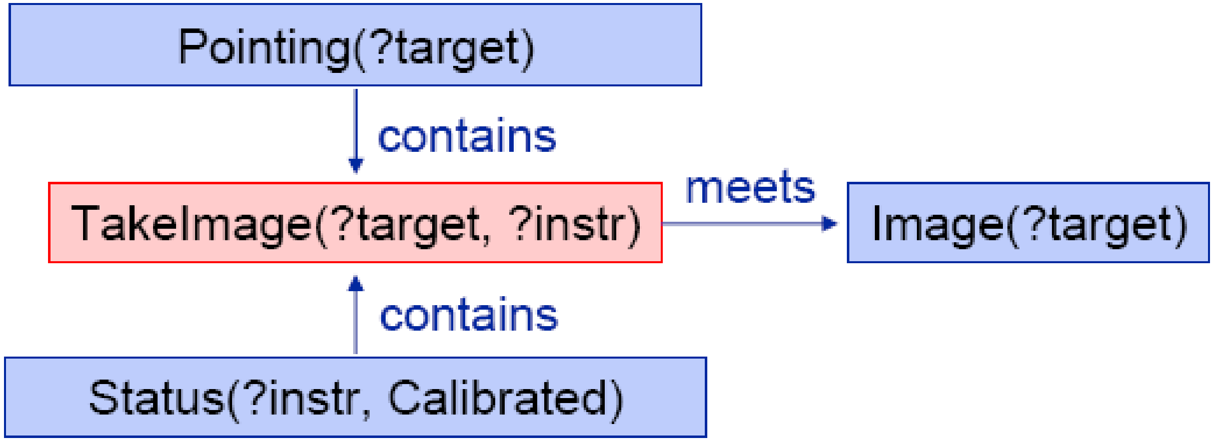
TakeImage (?target, ?instr):
  Pre: Status(?instr, Calibrated), Pointing(?target)
  Eff: Image(?target)

↓

TakeImage (?target, ?instr)	
contained-by	Status(?instr, Calibrated)
contained-by	Pointing(?target)
meets	Image(?target)

Temporal Operators

TakeImage (?target, ?instr)		
	contained-by	Status(?instr, Calibrated)
	contained-by	Pointing(?target)
	meets	Image(?target)



Based on slides by Dave Smith, NASA Ames

Temporal Operators

TakeImage (?target, ?instr)

contained-by	Status(?instr, Calibrated)
contained-by	Pointing(?target)
meets	Image(?target)

↓

$$\text{TakeImage(?target, ?instr)}_A$$

$$\Rightarrow \exists P\ \{\text{Status(?instr, Calibrated)}_P \wedge \text{Contains}(P, A)\}$$

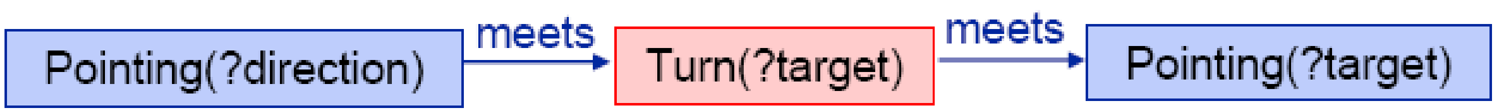
$$\wedge\ \exists Q\ \{\text{Pointing(?target)}_Q \wedge \text{Contains}(Q, A)\}$$

$$\wedge\ \exists R\ \{\text{Image(?target)}_R \wedge \text{Meets}(A, R)\}$$

Based on slides by Dave Smith, NASA Ames

Temporal Operators

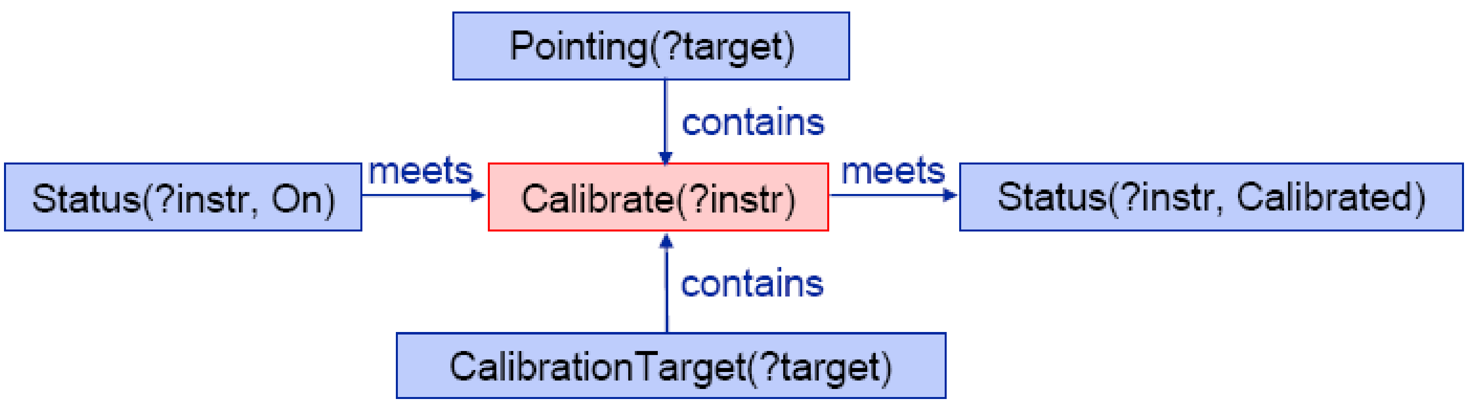
Turn (?target)
met-by Pointing(?direction)
meets Pointing(?target)



Temporal Operators

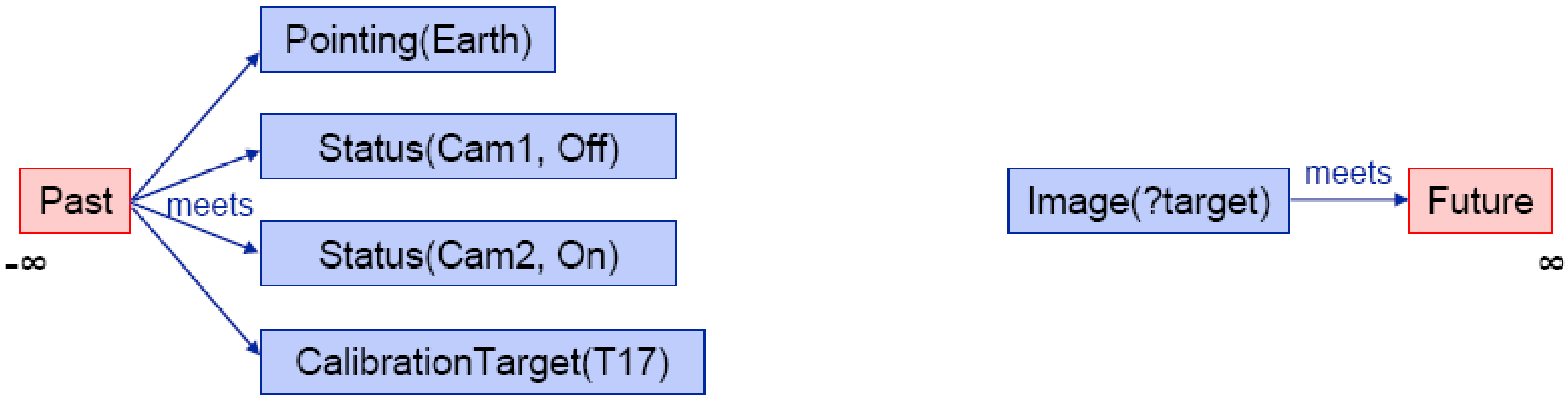
Calibrate (?instr)

met-by	Status(?instr, On)
contained-by	CalibrationTarget(?target)
contained-by	Pointing(?target)
meets	Status(?instr, Calibrated)



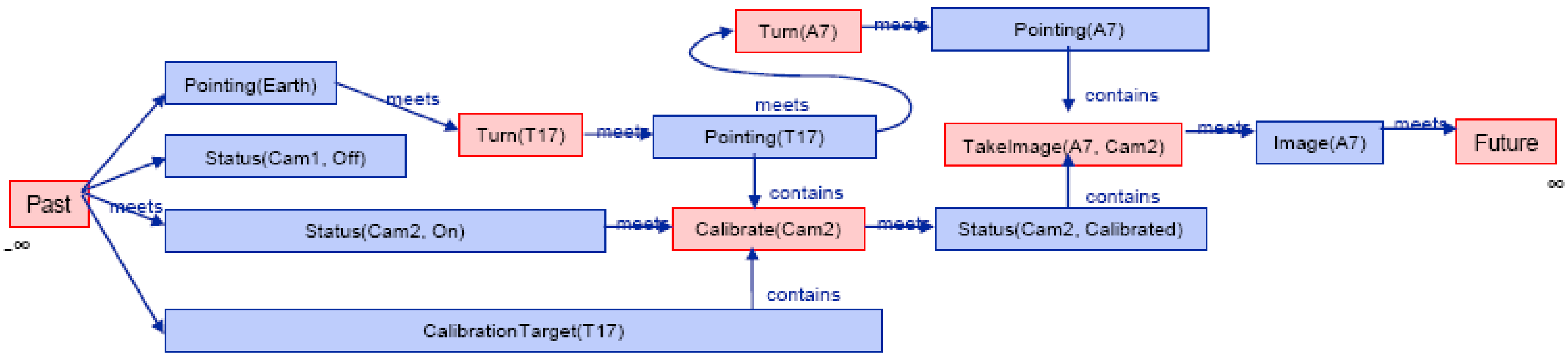
Based on slides by Dave Smith, NASA Ames

Temporal Planning Problem



Based on slides by Dave Smith, NASA Ames

Consistent Complete Plan



Based on slides by Dave Smith, NASA Ames

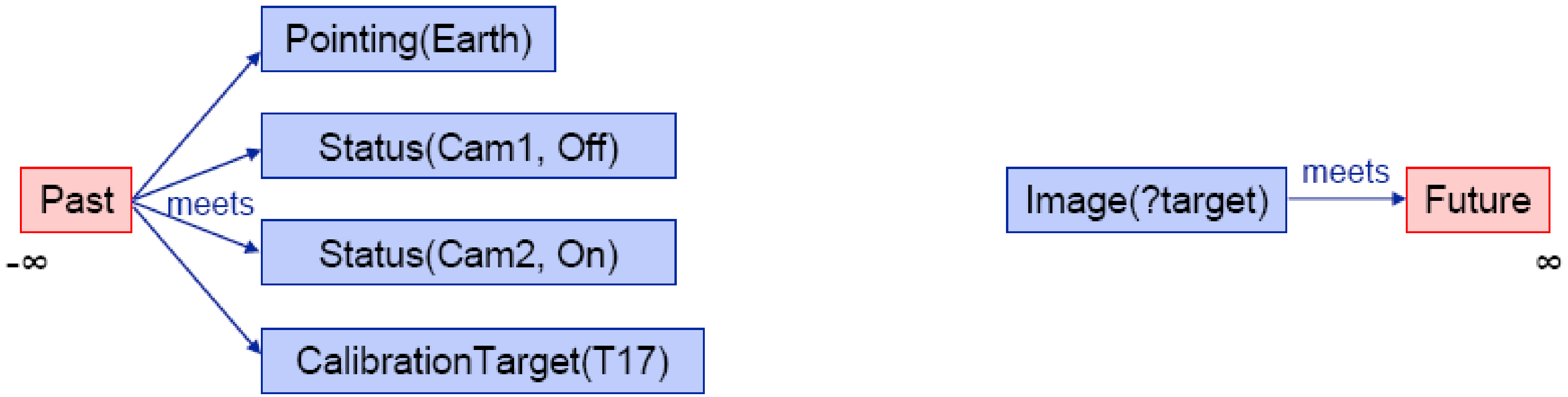
CBI-Planning

Choose:

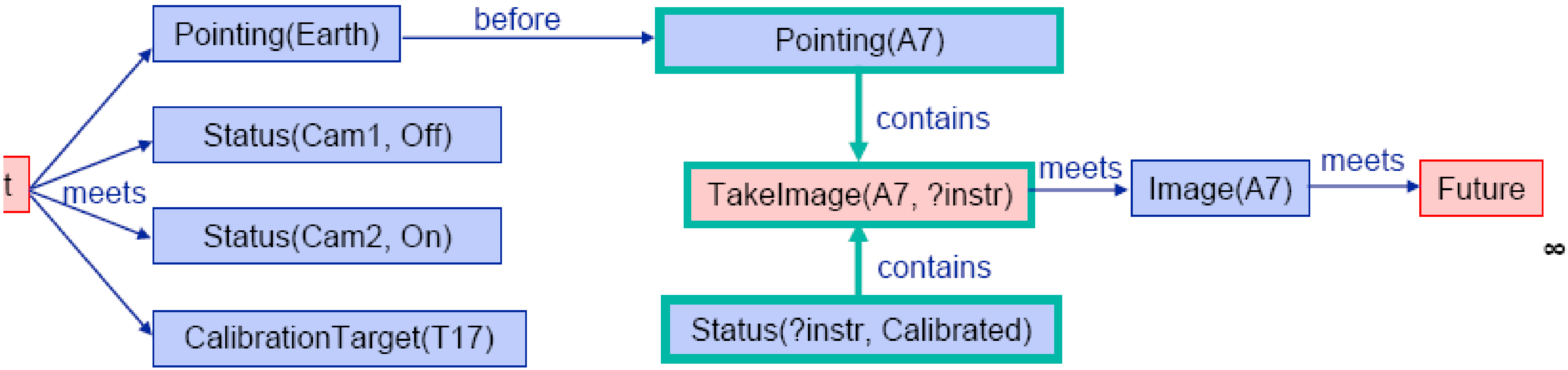
- introduce an action & instantiate constraints
- coalesce propositions

Propagate constraints

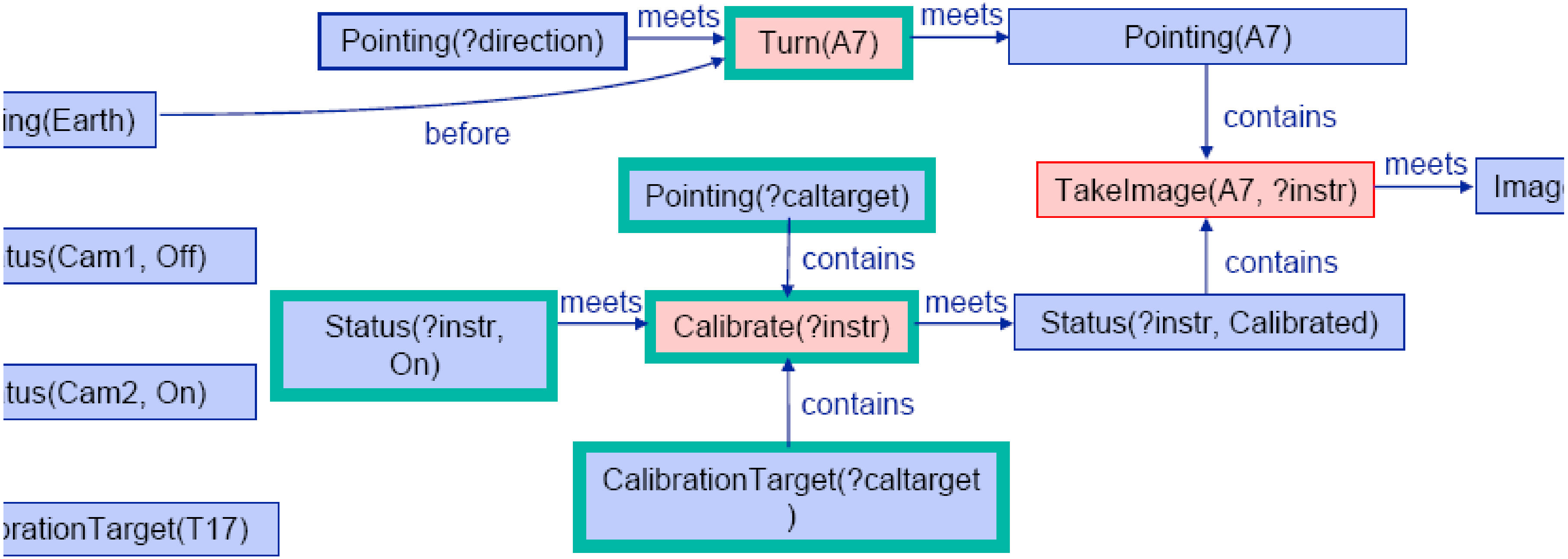
Initial Plan



Expansion

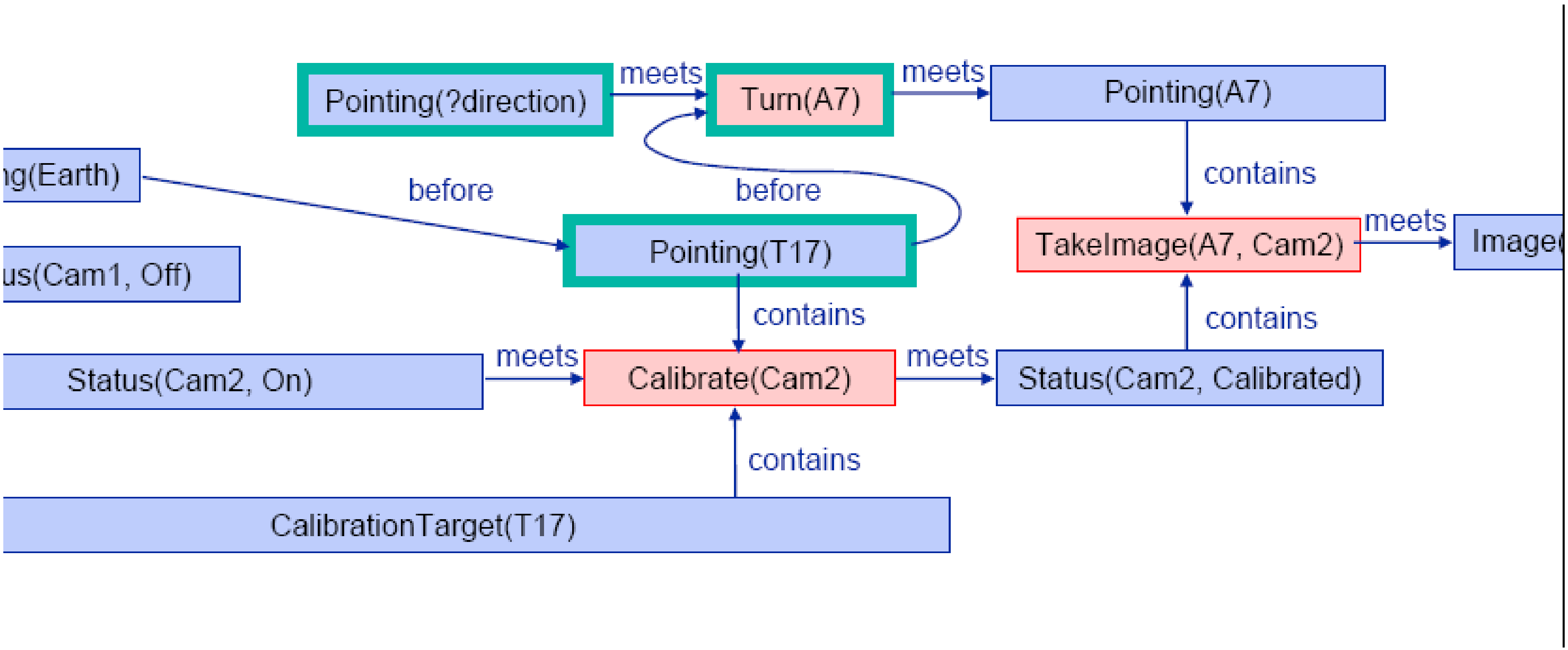

t
Pointing(Earth)
Status(Cam1, Off)
meets
Status(Cam2, On)
CalibrationTarget(T17)
before
Pointing(A7)
contains
TakeImage(A7, ?instr)
contains
Status(?instr, Calibrated)
meets
Image(A7)
meets
Future
∞

Expansion



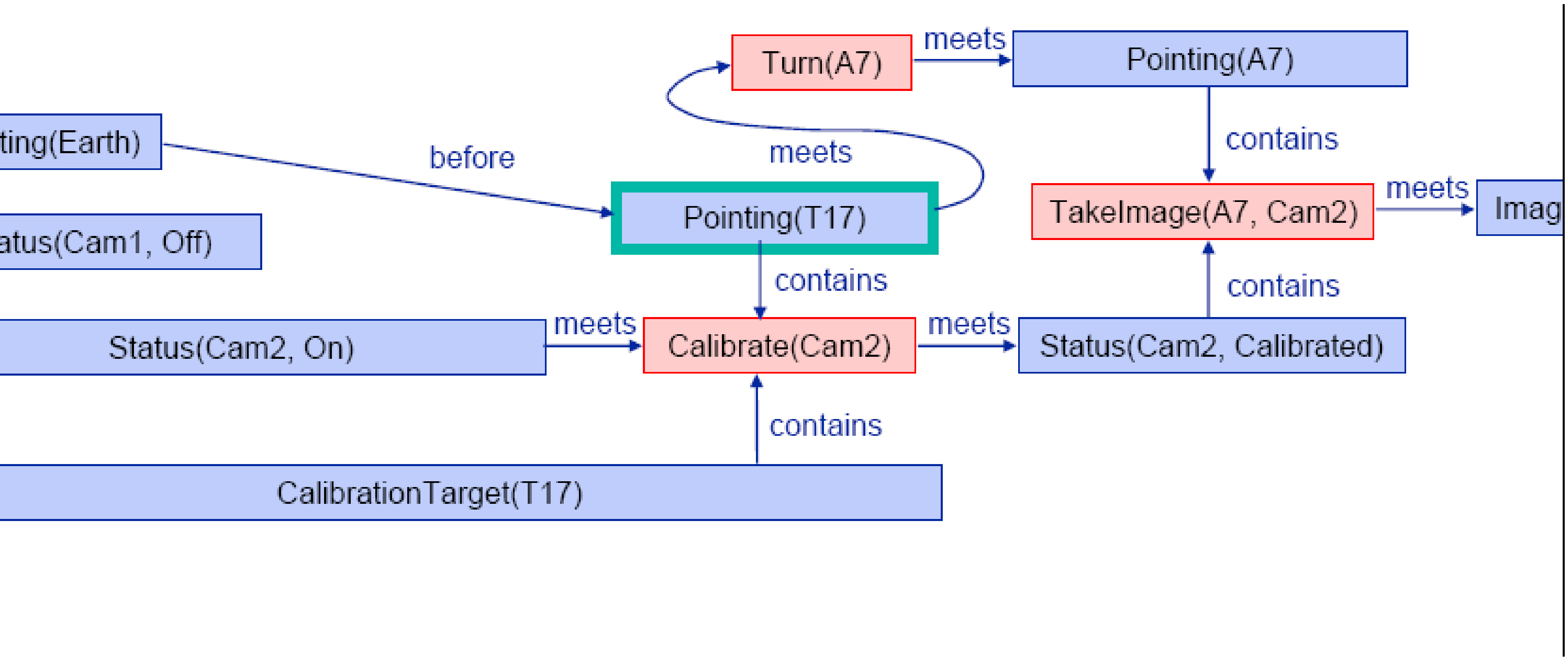
Based on slides by Dave Smith, NASA Ames

Coalescing



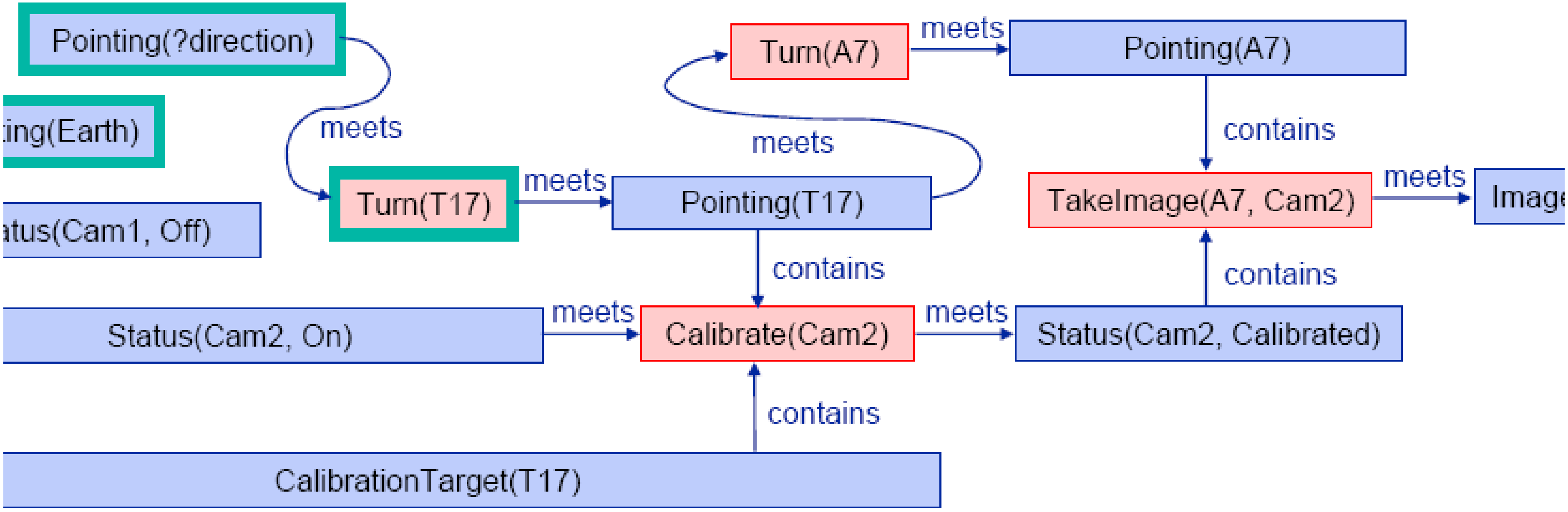
Pointing(?direction)
meets
Turn(A7)
meets
Pointing(A7)
ng(Earth)
before
before
contains
Pointing(T17)
TakeImage(A7, Cam2)
meets
Image(
us(Cam1, Off)
contains
contains
meets
meets
Status(Cam2, On)
Calibrate(Cam2)
Status(Cam2, Calibrated)
contains
CalibrationTarget(T17)

Coalescing



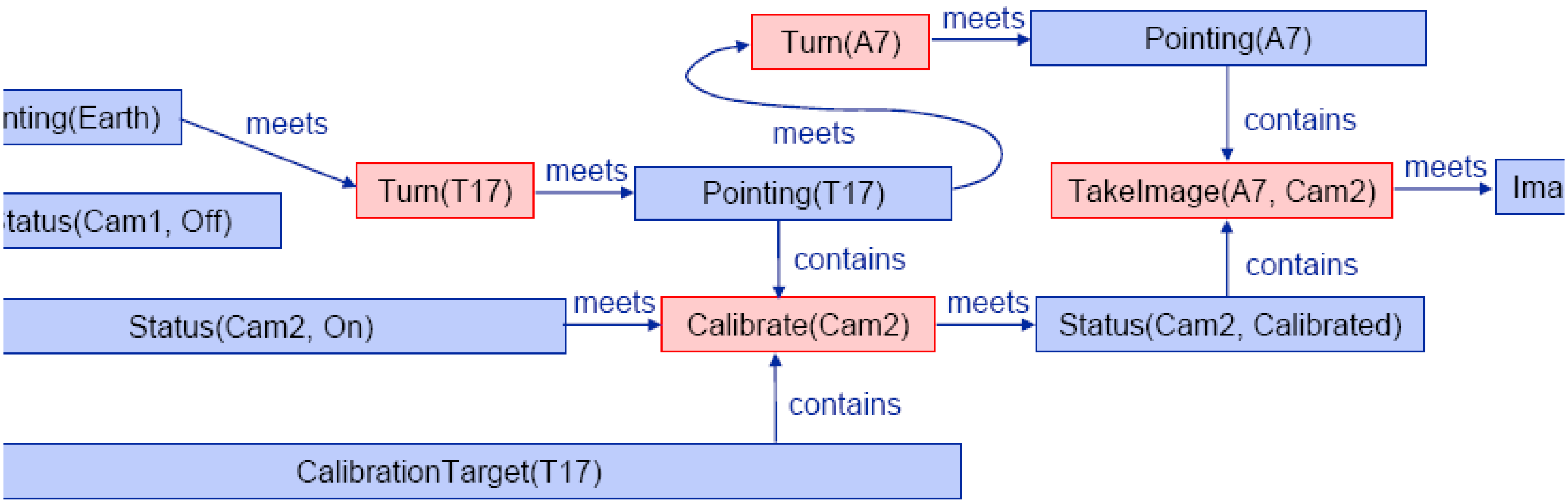
Turn(A7)
meets
Pointing(A7)
contains
ting(Earth)
before
meets
Pointing(T17)
TakeImage(A7, Cam2)
meets
Imag
atus(Cam1, Off)
contains
contains
Status(Cam2, On)
meets
Calibrate(Cam2)
meets
Status(Cam2, Calibrated)
contains
CalibrationTarget(T17)

Expansion



Pointing(?direction)
ting(Earth)
atus(Cam1, Off)
Turn(T17)
meets
meets
Pointing(T17)
Turn(A7)
meets
Pointing(A7)
meets
contains
TakeImage(A7, Cam2)
meets
Image
contains
Status(Cam2, On)
meets
Calibrate(Cam2)
meets
Status(Cam2, Calibrated)
contains
contains
CalibrationTarget(T17)

Coalescing



Turn(A7)
meets
Pointing(A7)
nting(Earth)
meets
meets
contains
Turn(T17)
meets
Pointing(T17)
TakeImage(A7, Cam2)
meets
Ima
tatus(Cam1, Off)
contains
contains
Status(Cam2, On)
meets
Calibrate(Cam2)
meets
Status(Cam2, Calibrated)
contains
CalibrationTarget(T17)

CBI-Algorithm

Expand(TQAs, constraints)

1. If the constraints are inconsistent, **fail**
2. If all TQAs have causal explanations, **return**(TQAs, constraints)
3. Select a $g \in$ TQAs with no causal explanation
4. **Choose**:

    Choose another $p \in$ TQAs such that g can be coalesced with p under constraints C

        Expand(TQAs-g, constraints $\cup$ C)

    Choose an action that would provide a causal explanation for g

        Let A be a new TQA for the action,
        and let R be the set of new TQAs implied by the axioms for A

        Let C be the constraints between A and R

        Expand(TQAs $\cup$ {A} $\cup$ R, constraints $\cup$ C)

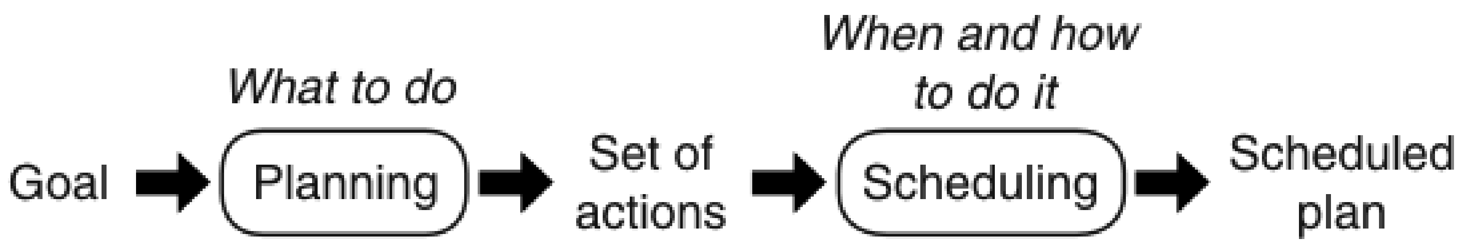
CBI-Planners

Zeno (Penberthy)	intervals, no CSP
Trains (Allen)	
Descartes (Joslin)	extreme least commitment
IxTeT (Ghallab)	functional rep.
HSTS (Muscettola)	functional rep., activities
EUROPA (Jonsson)	functional rep., activities

CBI vs POP

- CBI is similar to POP because least commitment and partial order
- But, temporal constraints in CBI …
- Contraints Temporal Network associated with a plan
- Constraint propagation

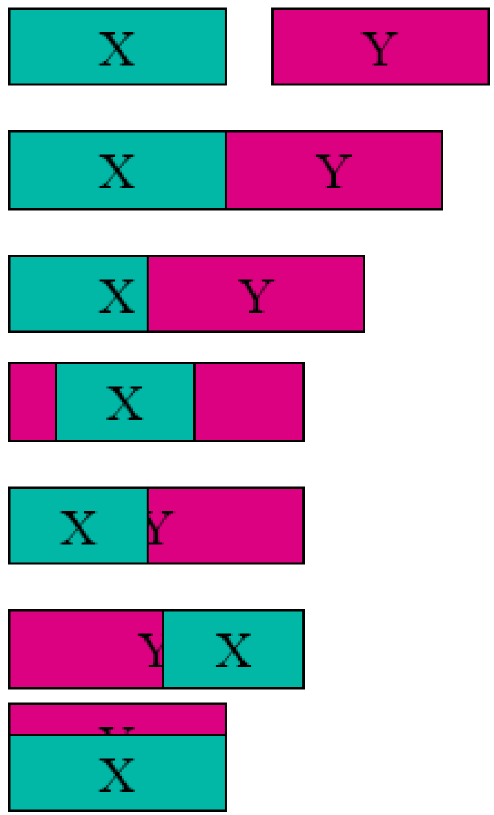
Planning and Scheduling



- Scheduling has usually been addressed separately from planning
- Thus, will give an overview of scheduling algorithms
- In some cases, cannot decompose planning and scheduling so cleanly

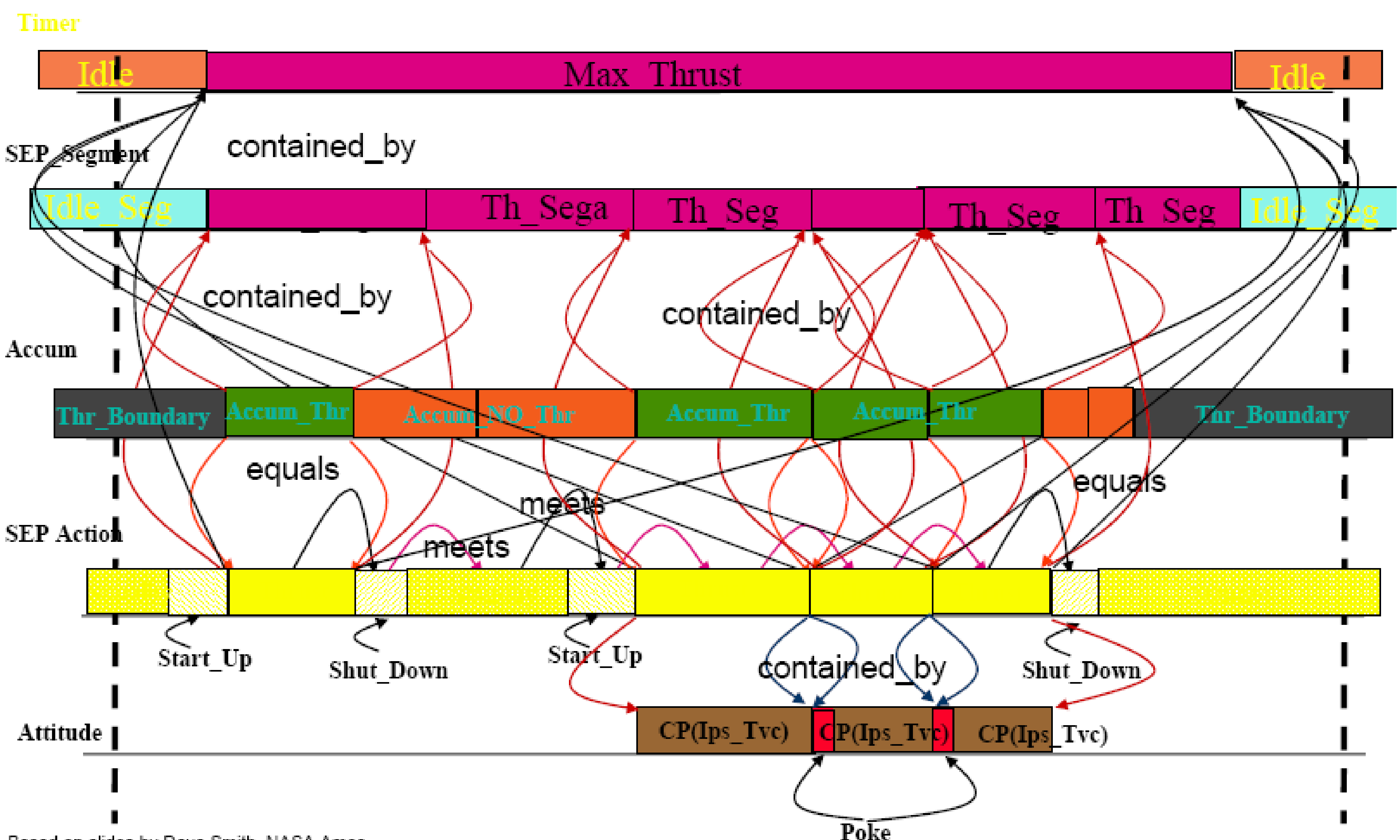
Temporal Constraints

- x before y
- x meets y
- x overlaps y
- x during y
- x starts y
- x finishes y
- x equals y



- y after x
- y met-by x
- y overlapped-by x
- y contains x
- y started-by x
- y finished-by x
- y equals x

RAX Example: DS1



Based on slides by Dave Smith, NASA Ames

Temporal Constraints as Inequalities

- x before y $X^{+} < Y^{-}$
- x meets y $X^{+} = Y^{-}$
- x overlaps y $(Y^{-} < X^{+})$ & $(X^{-} < Y^{+})$
- x during y $(Y^{-} < X^{-})$ & $(X^{+} < Y^{+})$
- x starts y $(X^{-} = Y^{-})$ & $(X^{+} < Y^{+})$
- x finishes y $(X^{-} < Y^{-})$ & $(X^{+} = Y^{+})$
- x equals y $(X^{-} = Y^{-})$ & $(X^{+} = Y^{+})$

Inequalities may be expressed as binary interval relations:

$$X^{+} - Y^{-} < [-inf, 0]$$

Metric Constraints

- Going to the store takes at least 10 minutes and at most 30 minutes.
  $\rightarrow$ $10 \leq [T^{+}(\text{store}) - T^{-}(\text{store})] \leq 30$

- Bread should be eaten within a day of baking.
  $\rightarrow$ $0 \leq [T^{+}(\text{baking}) - T^{-}(\text{eating})] \leq 1$ day

- Inequalities, $X^{+} < Y^{-}$, may be expressed as binary interval relations:
  $\rightarrow$ $-\inf < [X^{+} - Y^{-}] < 0$

Temporal Constraint Networks

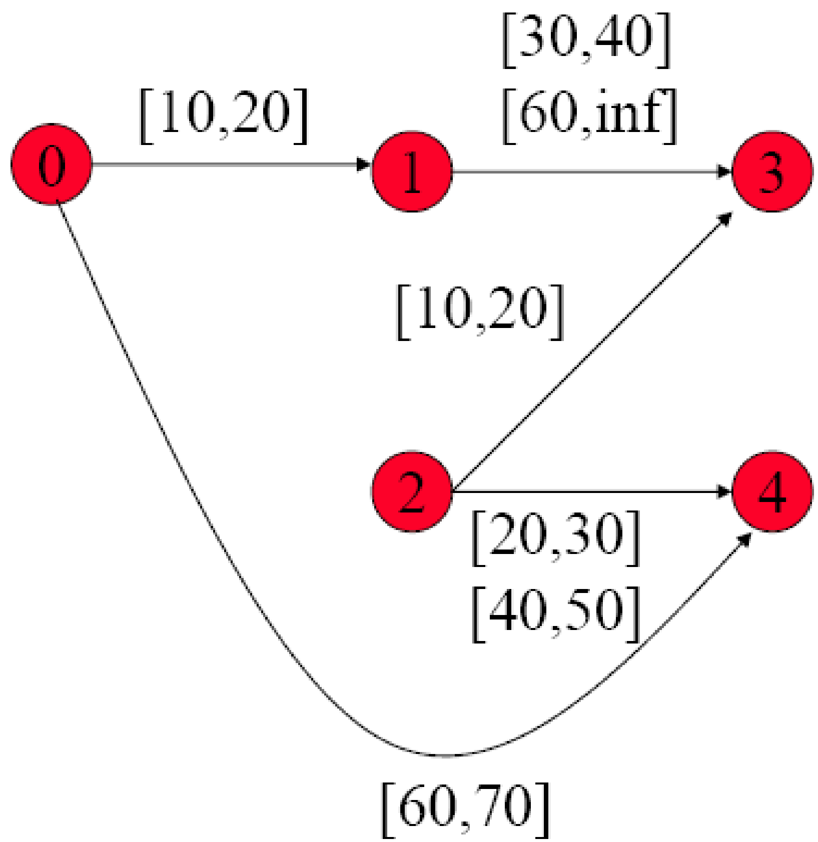
- A set of time points X_i at which events occur.
- Unary constraints

$$(a_0 \leq X_i \leq b_0) \text{ or } (a_1 \leq X_i \leq b_1) \text{ or } \ldots$$

- Binary constraints

$$(a_0 \leq X_j - X_i \leq b_0) \text{ or } (a_1 \leq X_j - X_i \leq b_1) \text{ or } \ldots$$

Temporal Constraint Satisfaction Problem



Simple Temporal Networks

Simple Temporal Networks:

- A set of time points X_i at which events occur.
- Unary constraints

  $(a_0 \leq X_i \leq b_0)$ ~~or $(a_1 \leq X_i \leq b_1)$ or . . .~~
- Binary constraints

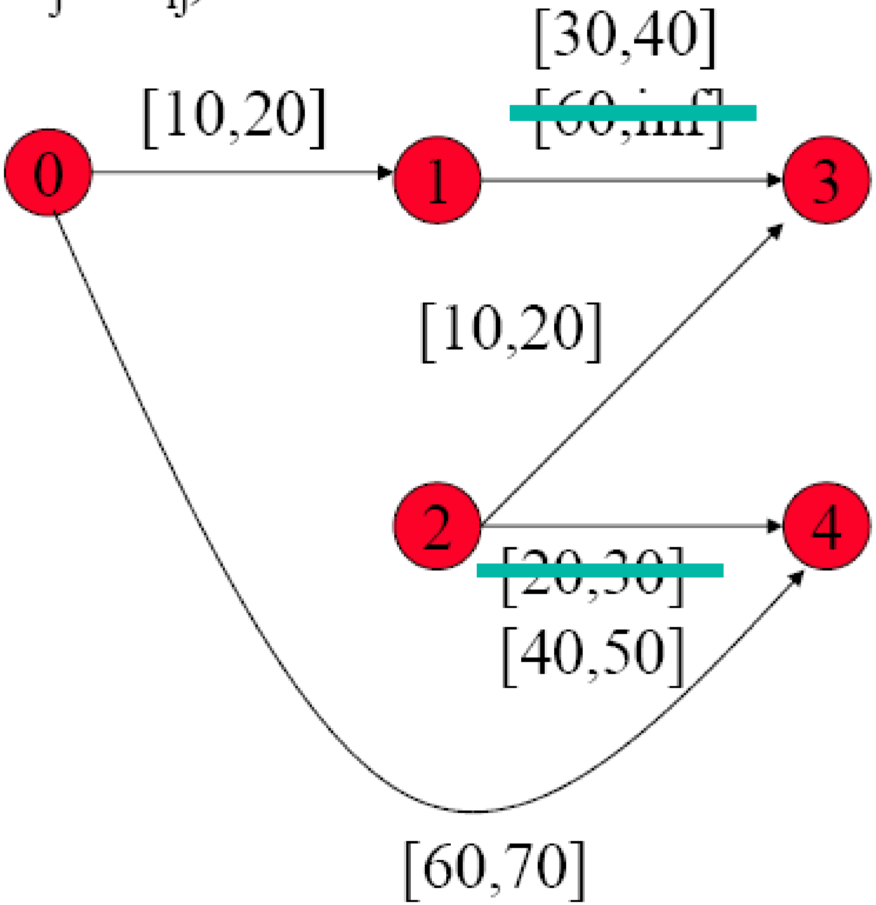
  $(a_0 \leq X_j - X_i \leq b_0)$ ~~or $(a_1 \leq X_j - X_i \leq b_1)$ or . . .~~

Sufficient to represent:
- most Allen relations
- simple metric constraints

Can't represent:
- Disjoint activities

Simple Temporal Networks

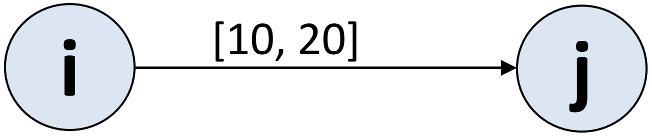
- $T_{ij} = (a_{ij} \leq X_i - X_j \leq b_{ij})$



Based on slides by Dave Smith, NASA Ames

Simple Temporal Network (STP)

- A special class of temporal problems
- Can be solved in polynomial time
- An edge e_{ij}: $i \rightarrow j$ is labeled by a **single** interval $[a_{ij}, b_{ij}]$



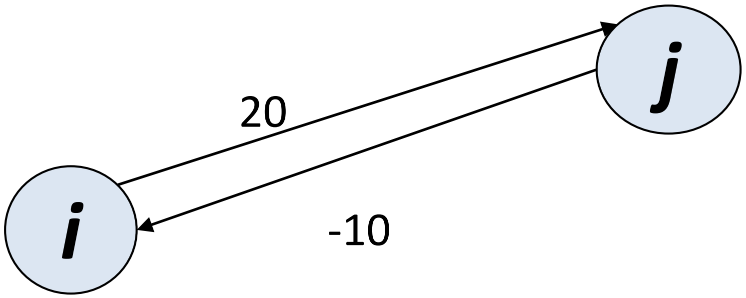
- Constraint ($a_{ij} \leq x_j - x_i \leq b_{ij}$) expressed by

$$(x_j - x_i \leq b_{ij}) \wedge (x_i - x_j \leq -a_{ij})$$

- Example $(x_j - x_i \leq 20) \wedge (x_i - x_j \leq -10)$

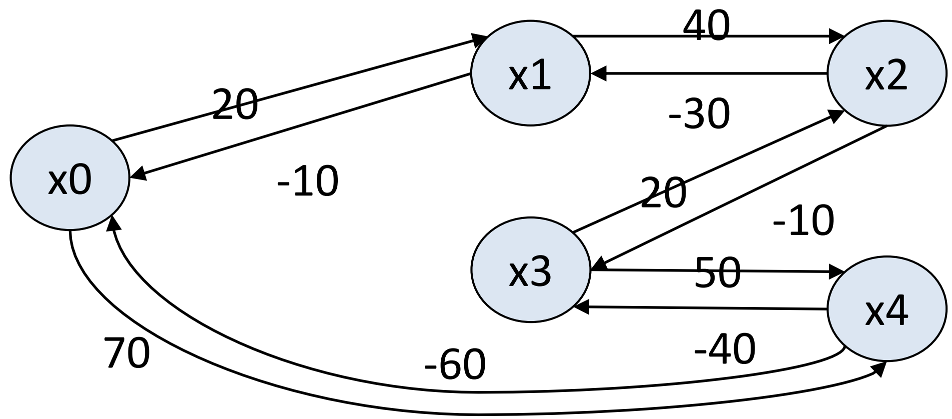
Distance Graph of an STP

- The STP is transformed into an all-pairs-shortest-paths problem on a **distance graph**
- Each constraint is replaced by two edges: one + and one -



- Constraint graph $\rightarrow$ directed cyclic graph

Solving the Distance Graph of the STP

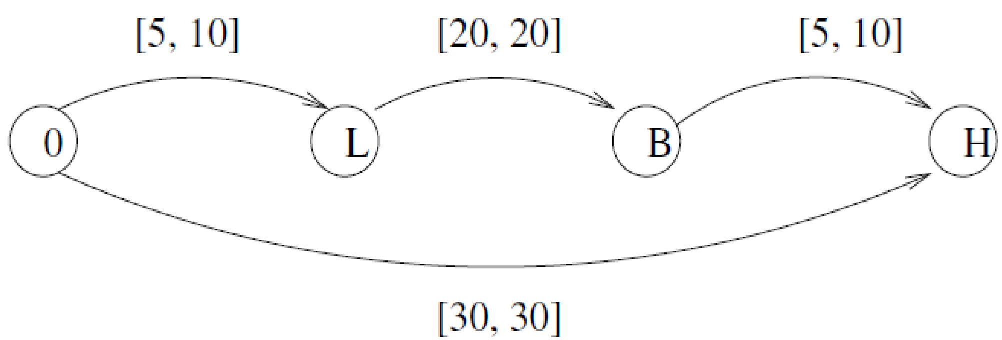


- Run **Floyd-Warshall** all pairs shortest path
- If any pair of nodes has a negative cycle $\Rightarrow$ inconsistency
- If consistent after **F-W** $\Rightarrow$ minimal & decomposable
- Once d-graph formed, assembling a solution by checking against the previous labeling
- Total time: F-W $O(n^3)$ + Assembling $O(n^2)$ = $O(n^3)$.

Example

- Eventi:

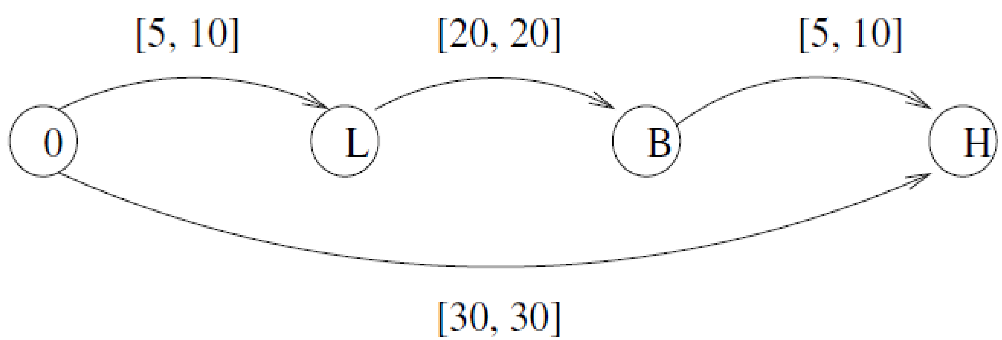
1. I was in Houghton at 8:30.
2. I left home between 8:05 and 8:10.
3. It takes me 20 minutes to drive to the bridge.
4. I waited 5-10 minutes at the bridge.



Example

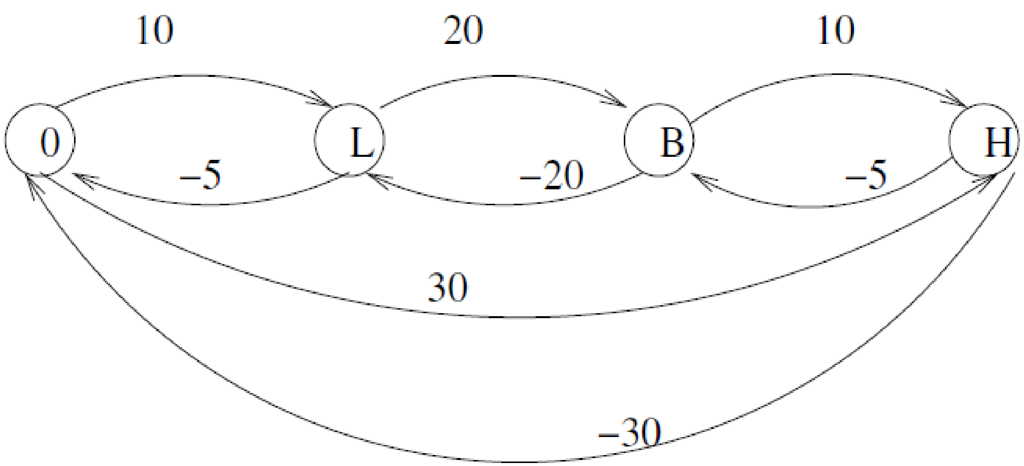
- Eventi:

1. I was in Houghton at 8:30.
2. I left home between 8:05 and 8:10.
3. It takes me 20 minutes to drive to the bridge.
4. I waited 5-10 minutes at the bridge.



Example

- Eventi:
  1. I was in Houghton at 8:30.
  2. I left home between 8:05 and 8:10.
  3. It takes me 20 minutes to drive to the bridge.
  4. I waited 5-10 minutes at the bridge.

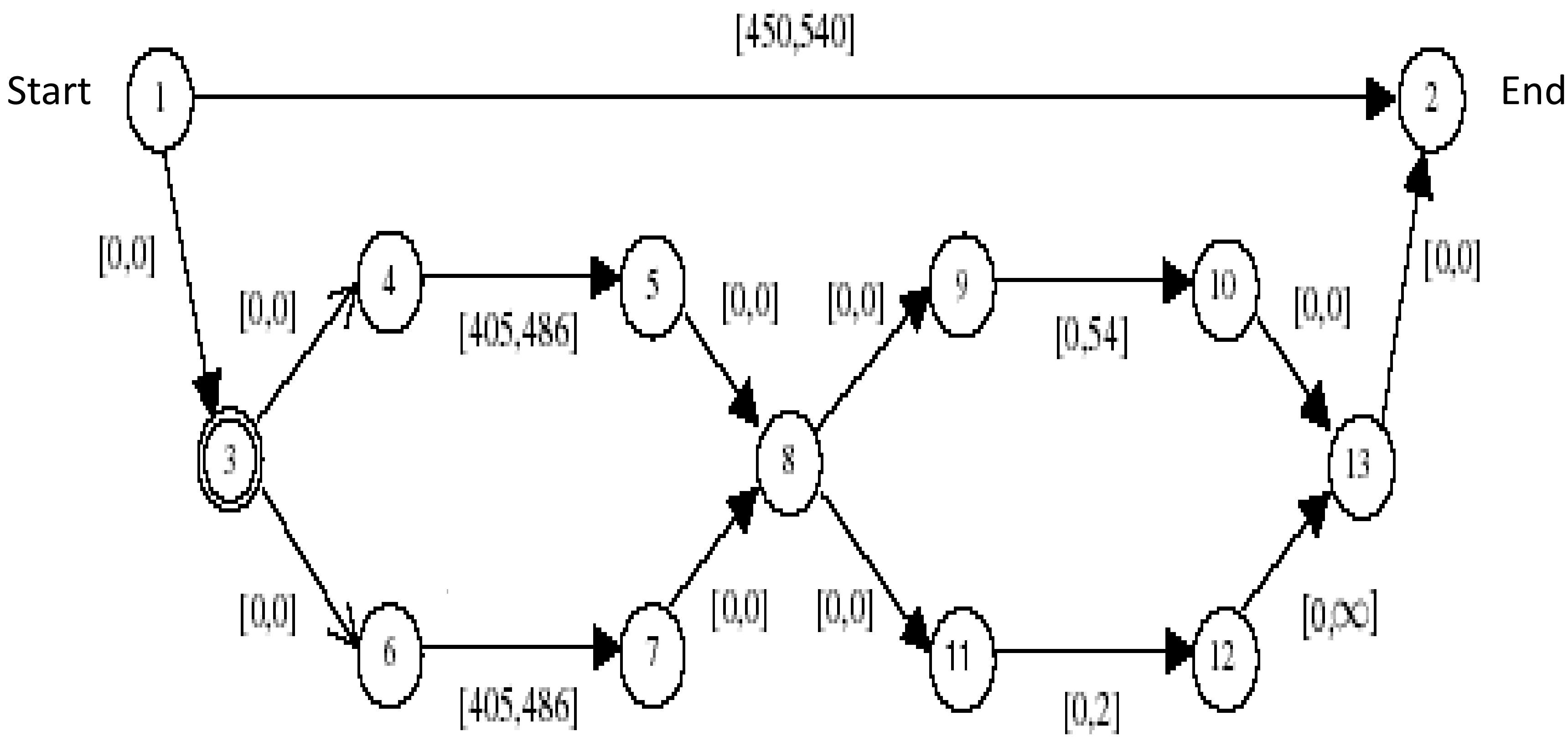


	0	1	2	3
0	0	10	99	30
1	-5	0	20	99
2	99	-20	0	10
3	-30	99	-5	0

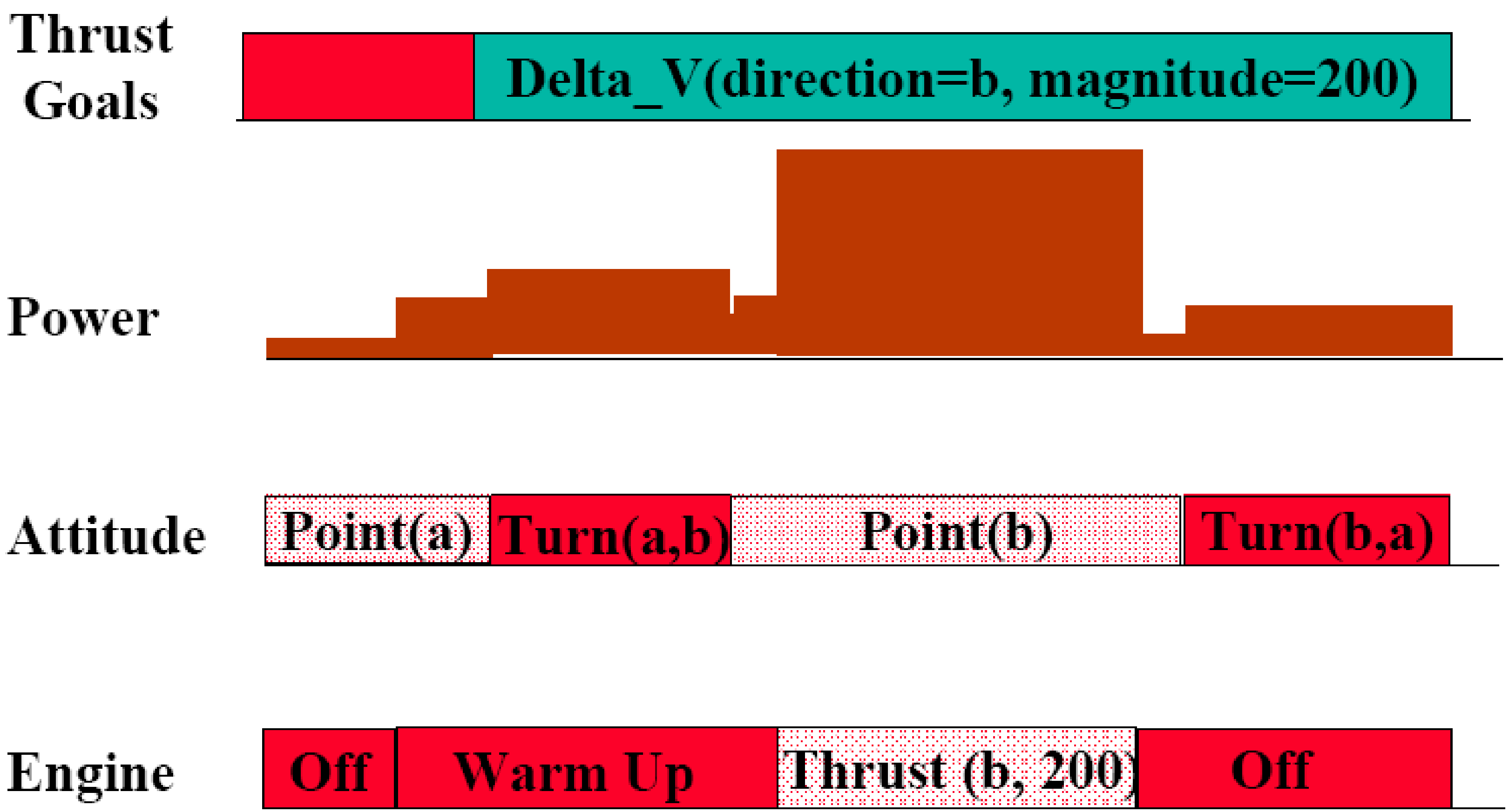
Floyd-Warshall

	0	1	2	3
0	0	**5**	25	30
1	-5	0	20	25
2	-25	-20	0	**5**
3	-30	-25	-5	0

STN example

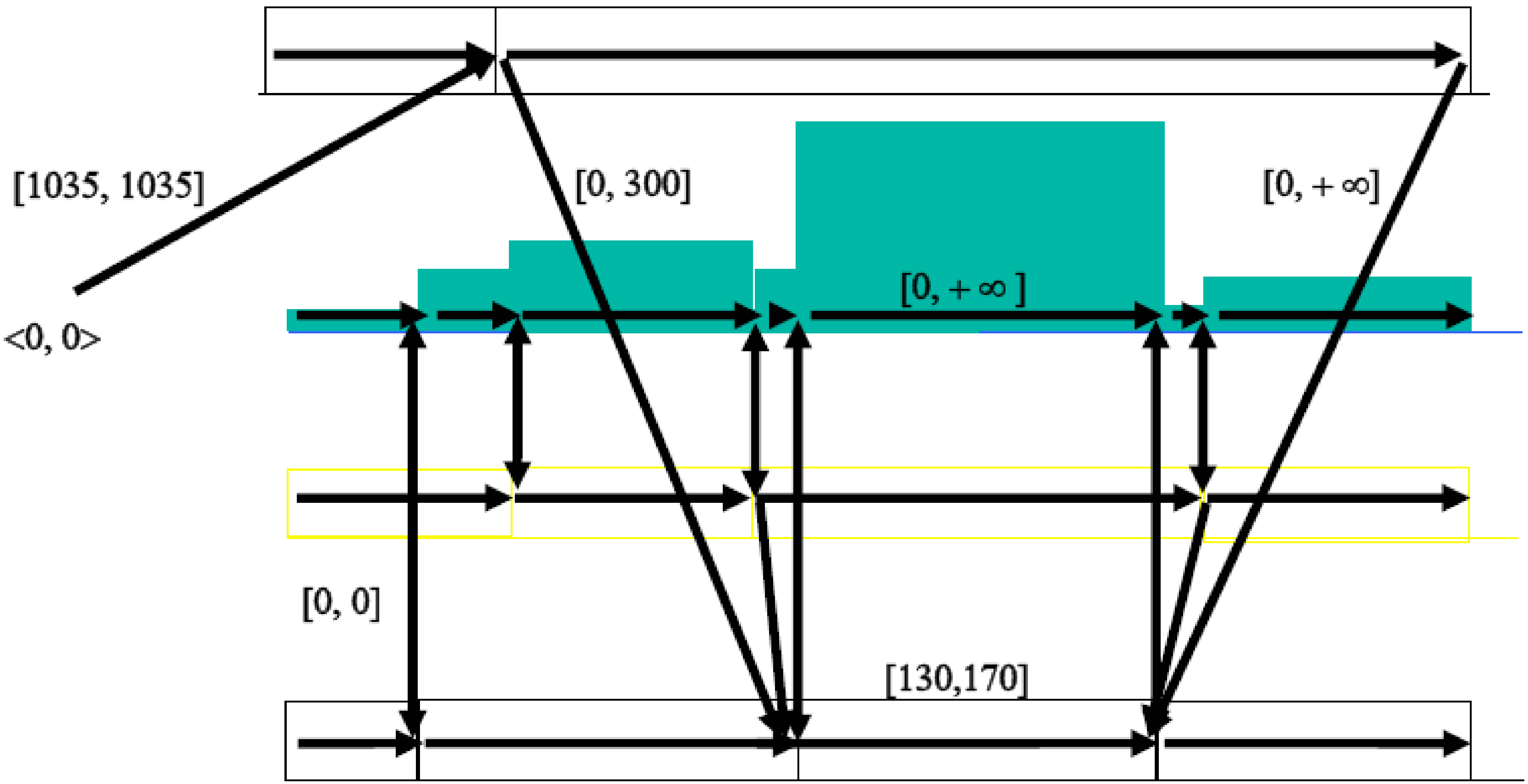

[450,540]
Start
1
2
End
[0,0]
3
[0,0]
4
[405,486]
5
[0,0]
[0,0]
8
9
[0,54]
10
[0,0]
13
[0,0]
[0,0]
6
[405,486]
7
[0,0]
[0,0]
11
[0,2]
12
[0,∞]

A Complete CBI-Plan is a STN



Based on slides by Dave Smith, NASA Ames

A Complete CBI-Plan is a STN



Based on slides by Dave Smith, NASA Ames

DS1: Remote Agent

Remote Agent Experiment: RAX

Remote Agent Experiment

See rax.arc.nasa.gov

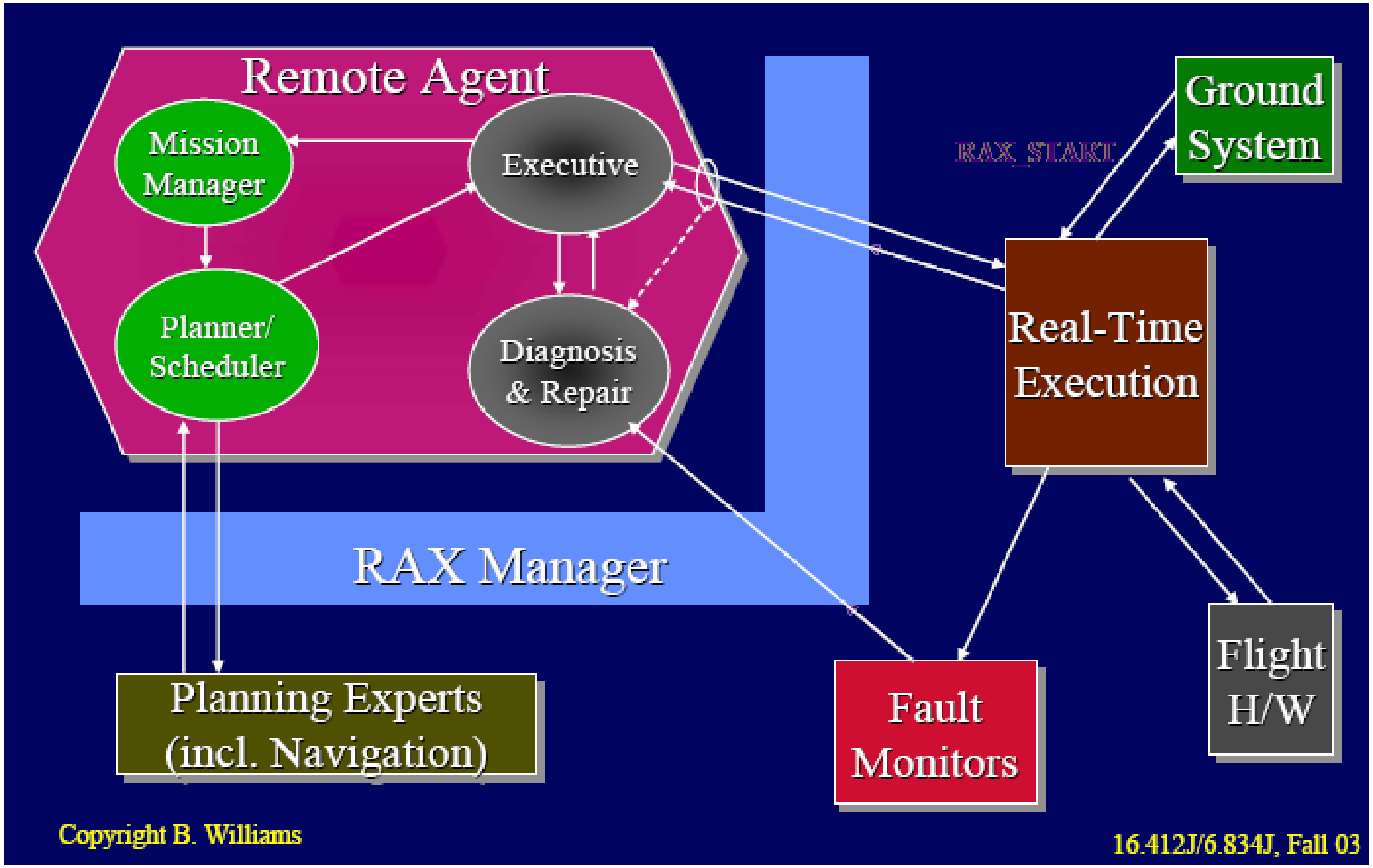
May 17-18th experiment

- Generate plan for course correction and thrust
- Diagnose camera as stuck on
  - Power constraints violated, abort current plan and replan
- Perform optical navigation
- Perform ion propulsion thrust

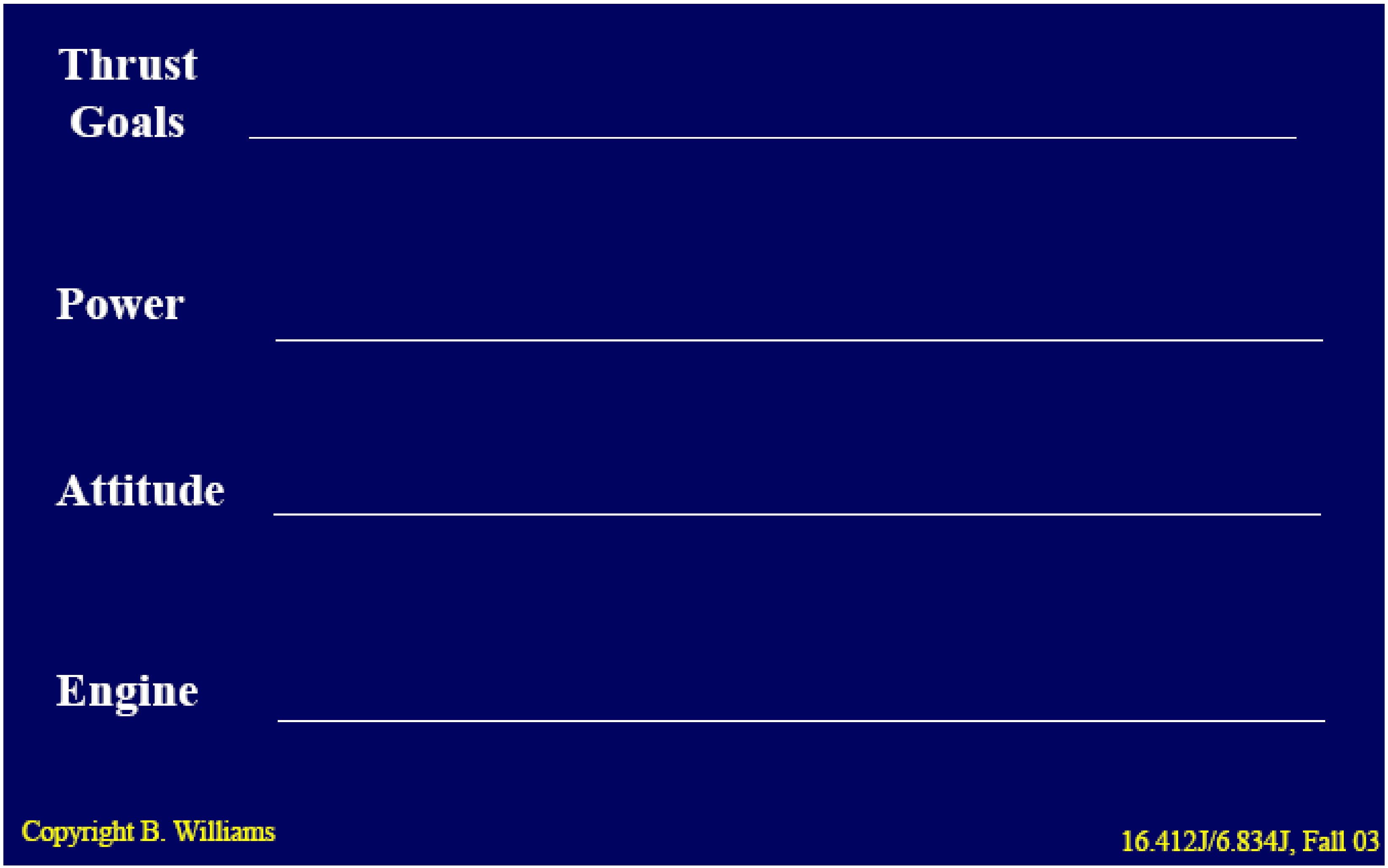
May 21th experiment.

- Diagnose faulty device and
  - Repair by issuing reset.
- Diagnose switch sensor failure.
  - Determine harmless, and continue plan.
- Diagnose thruster stuck closed and
  - Repair by switching to alternate method of thrusting.
- Back to back planning

Remote Agent

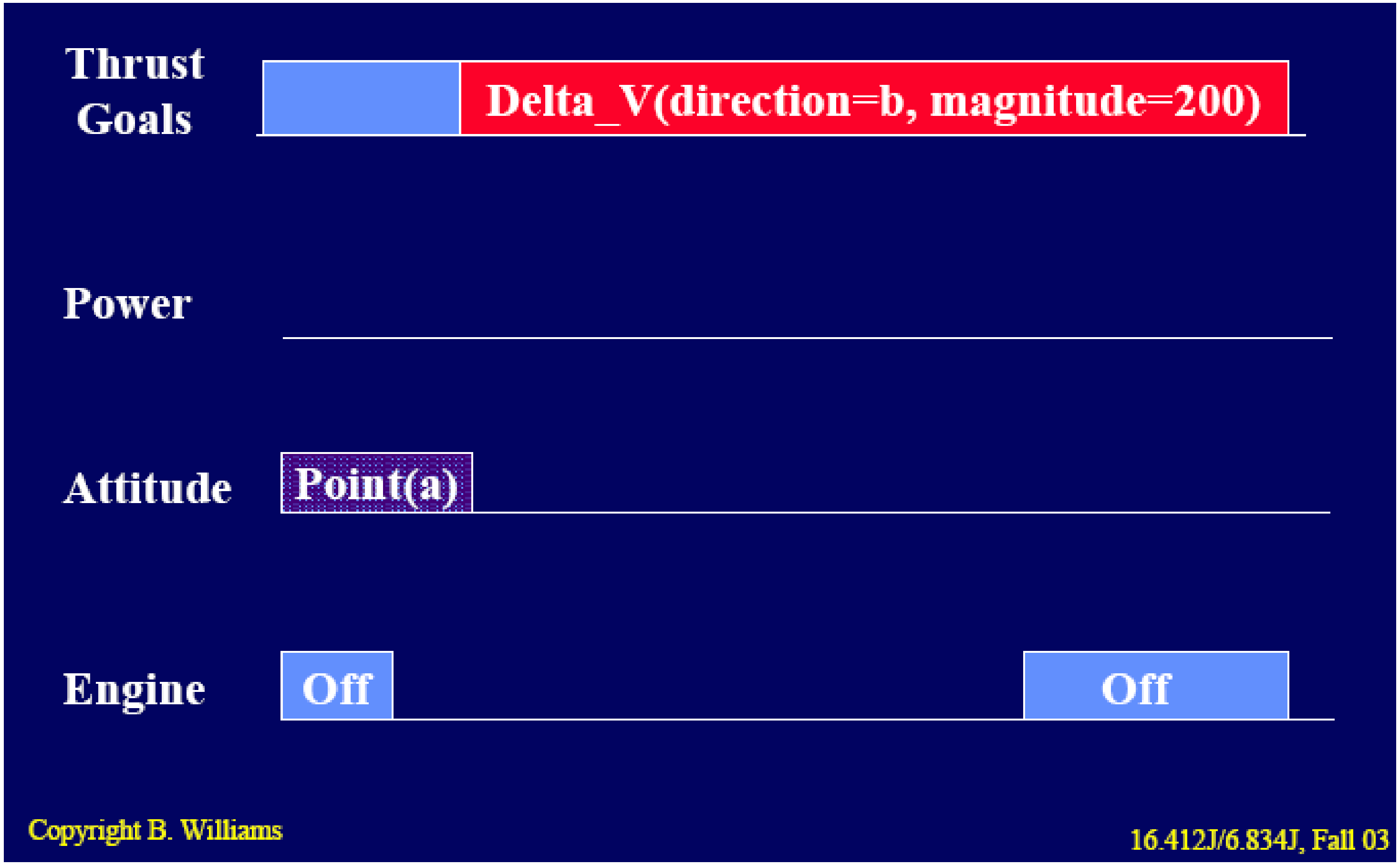


Remote Agent



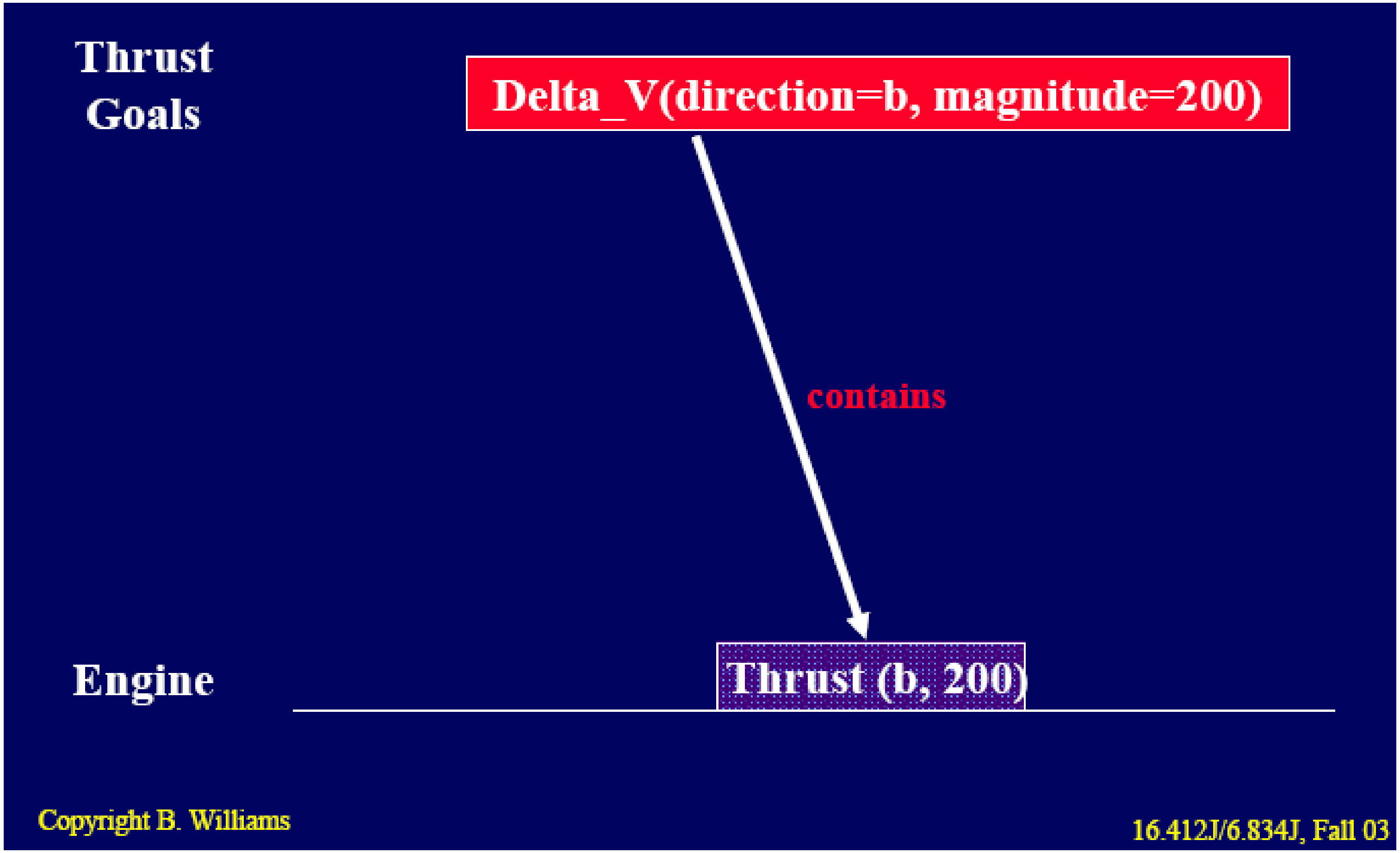
Remote Agent

- Mission Manager



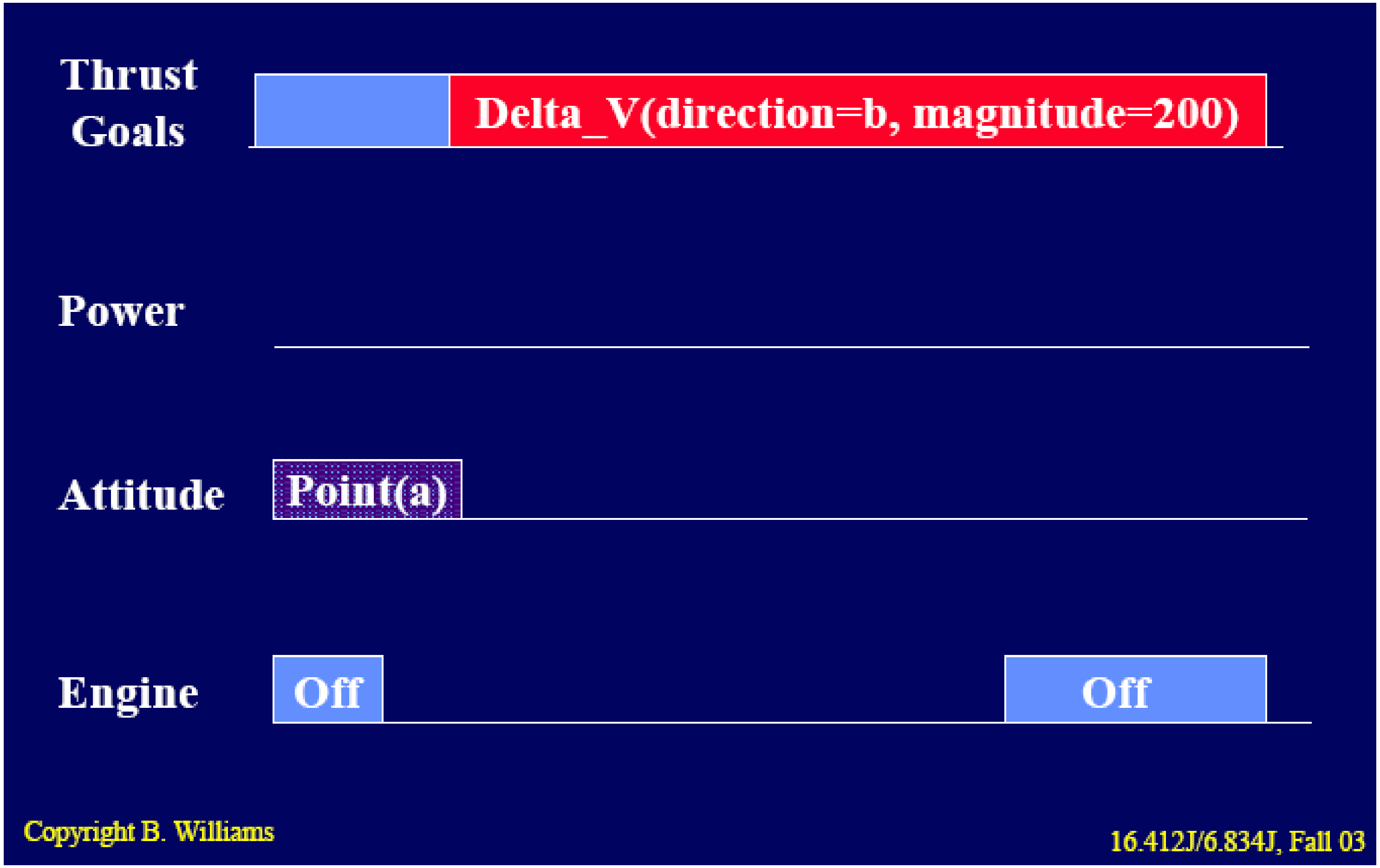
Remote Agent

- Constraints:



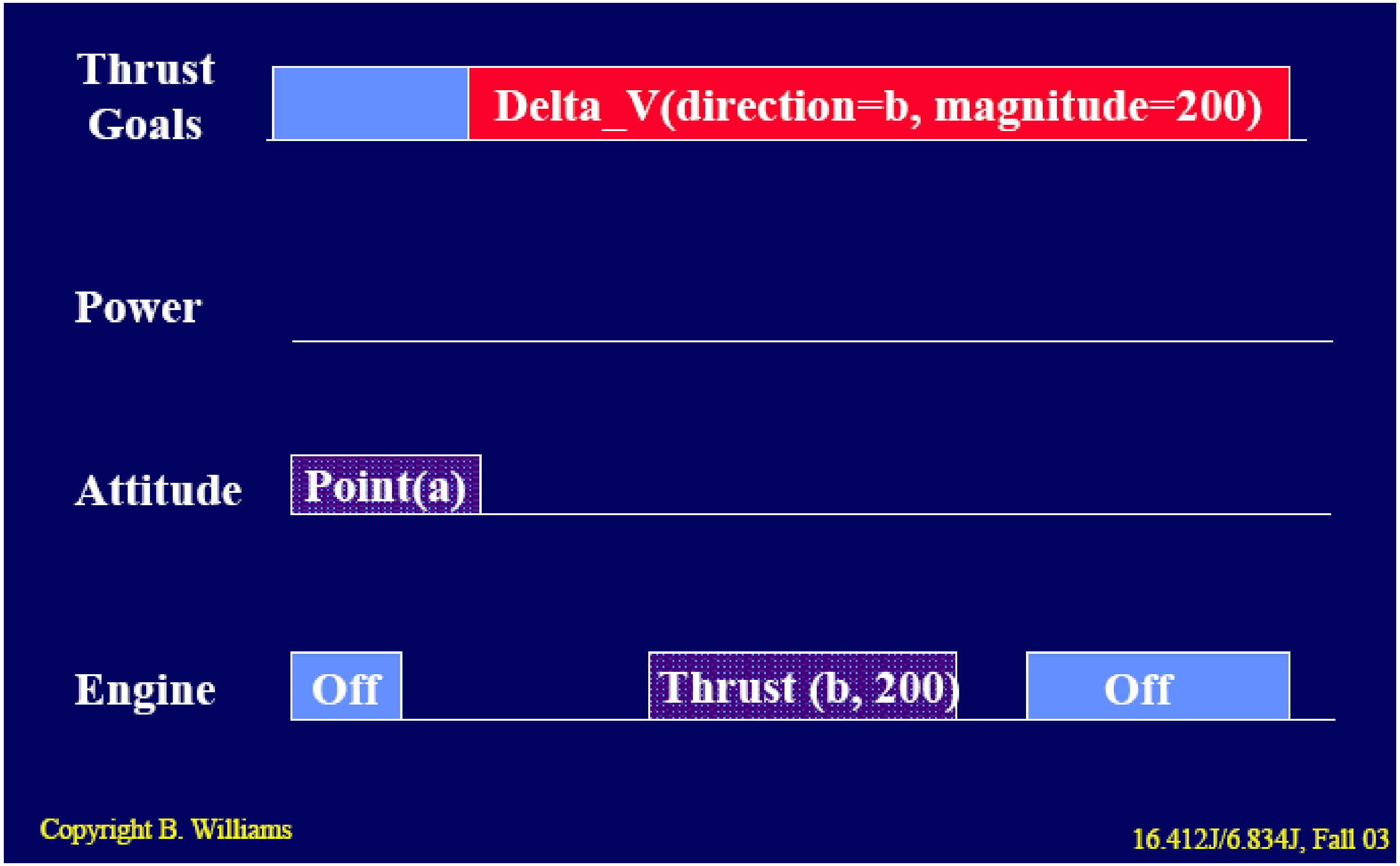
Remote Agent

- Planner starts



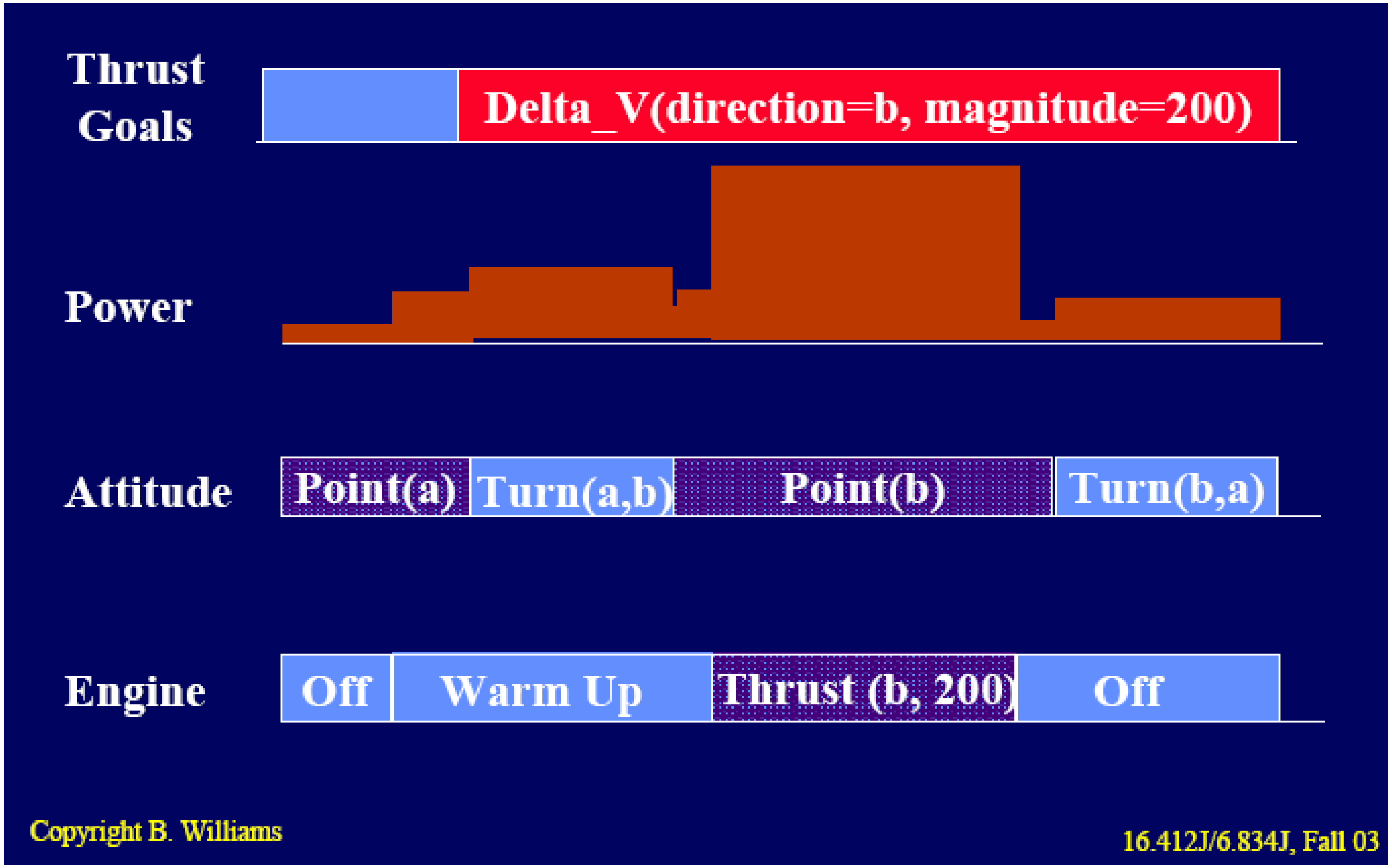
Remote Agent

- Planning



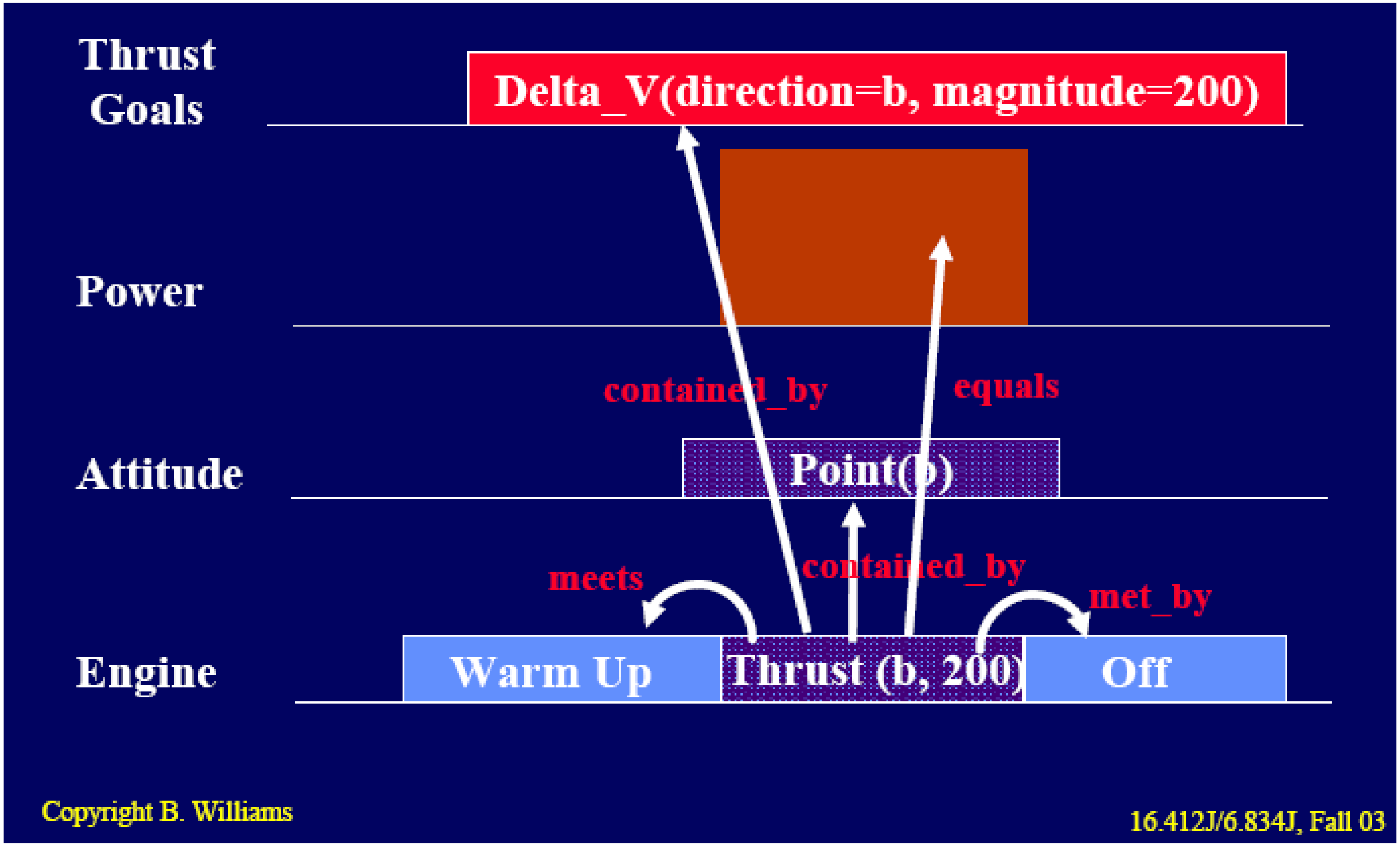
Remote Agent

- Final Plan



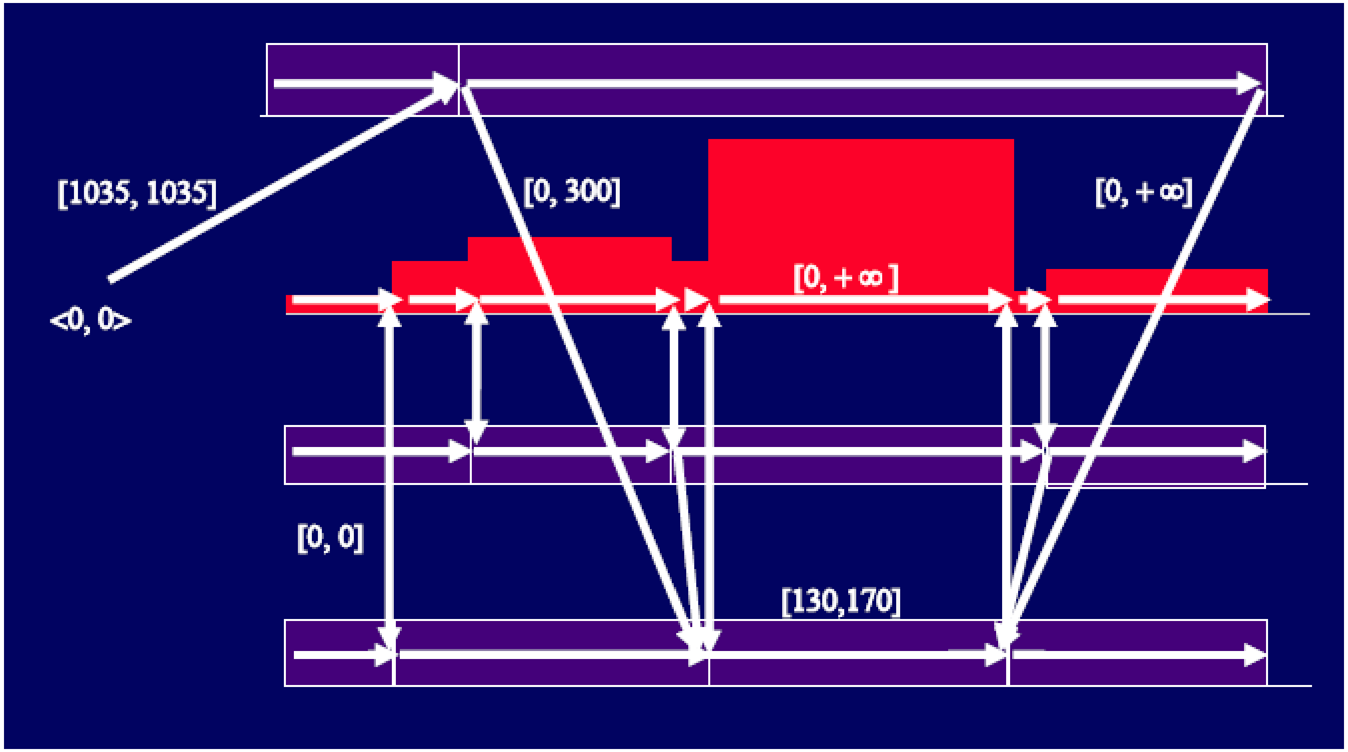
Remote Agent

- Constraints



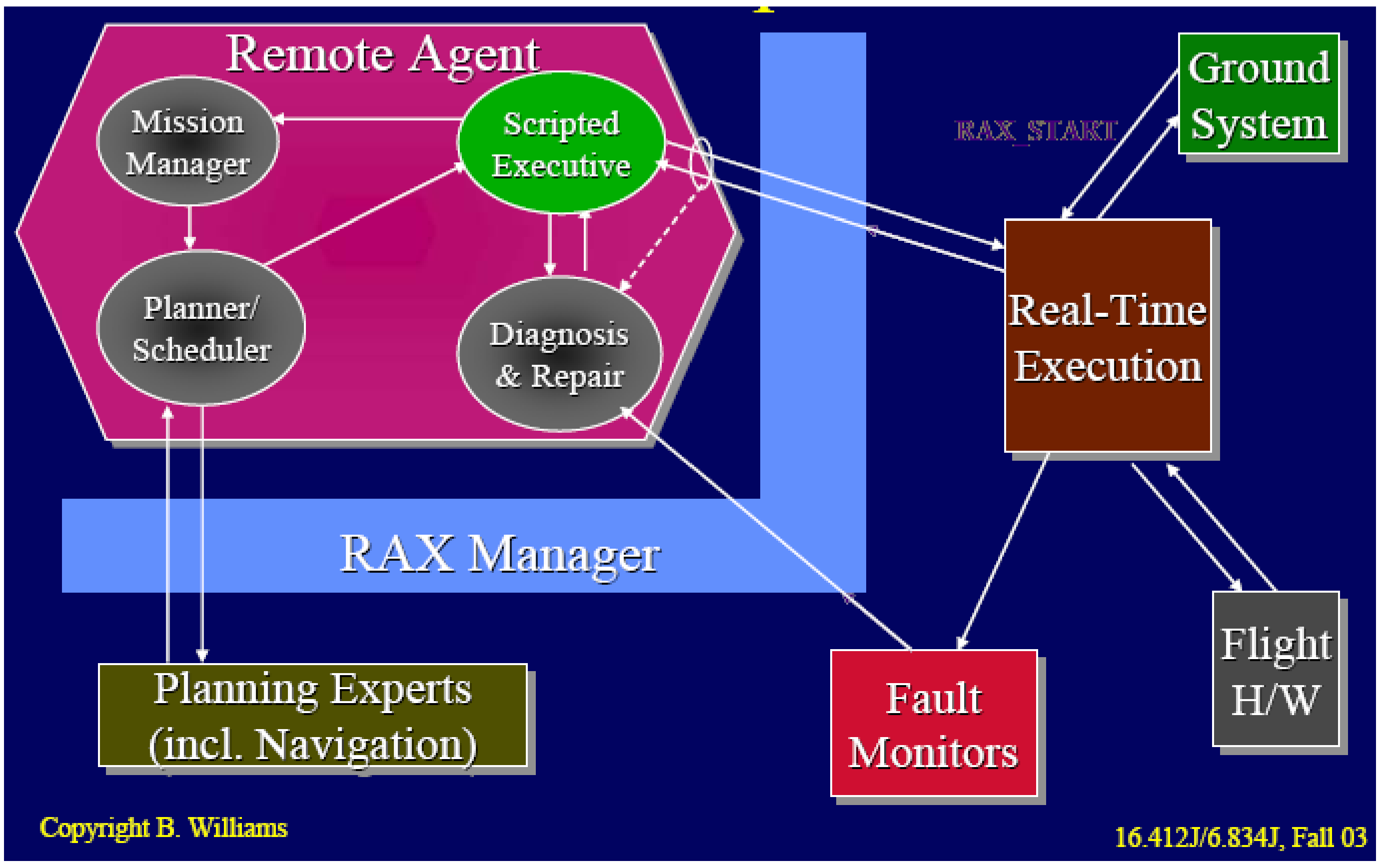
Remote Agent

- Flexible Temporal Plan through least commitment



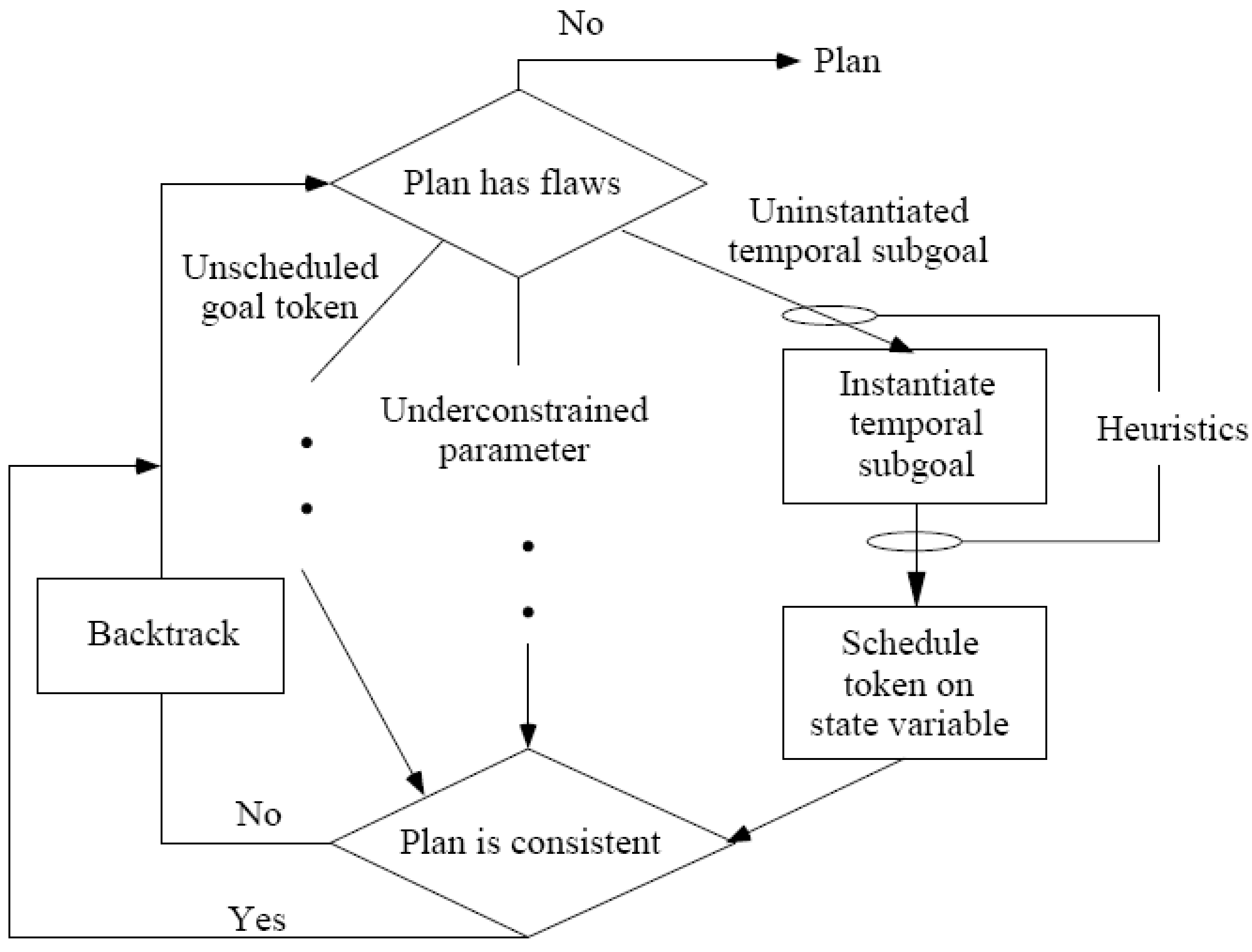
Remote Agent

- Executive system dispatch tasks



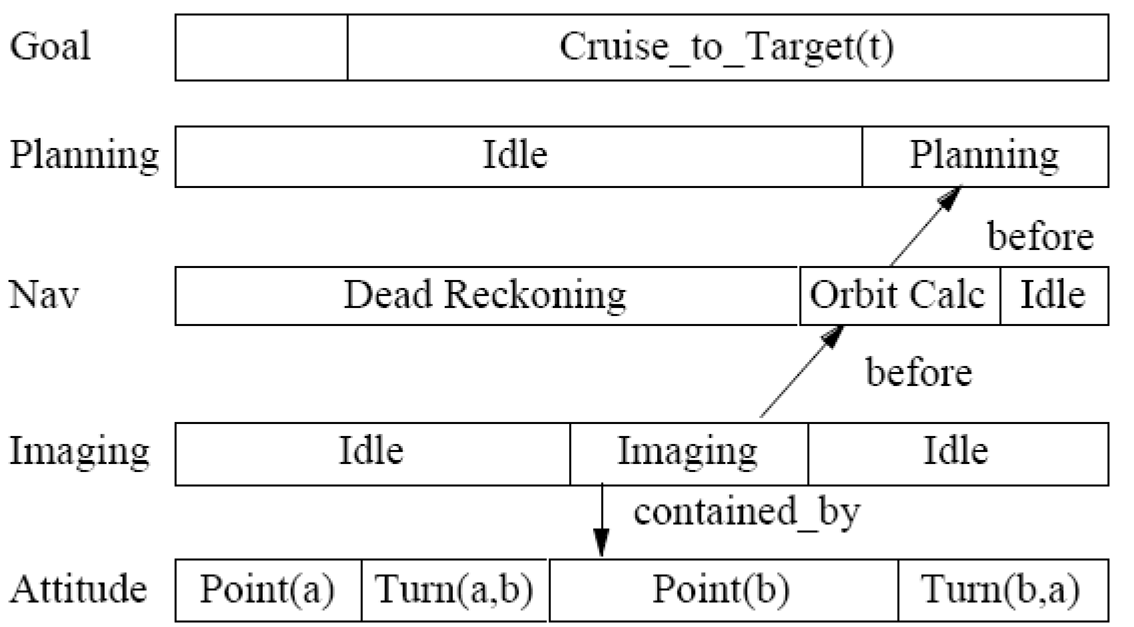
Remote Agent

- Planning



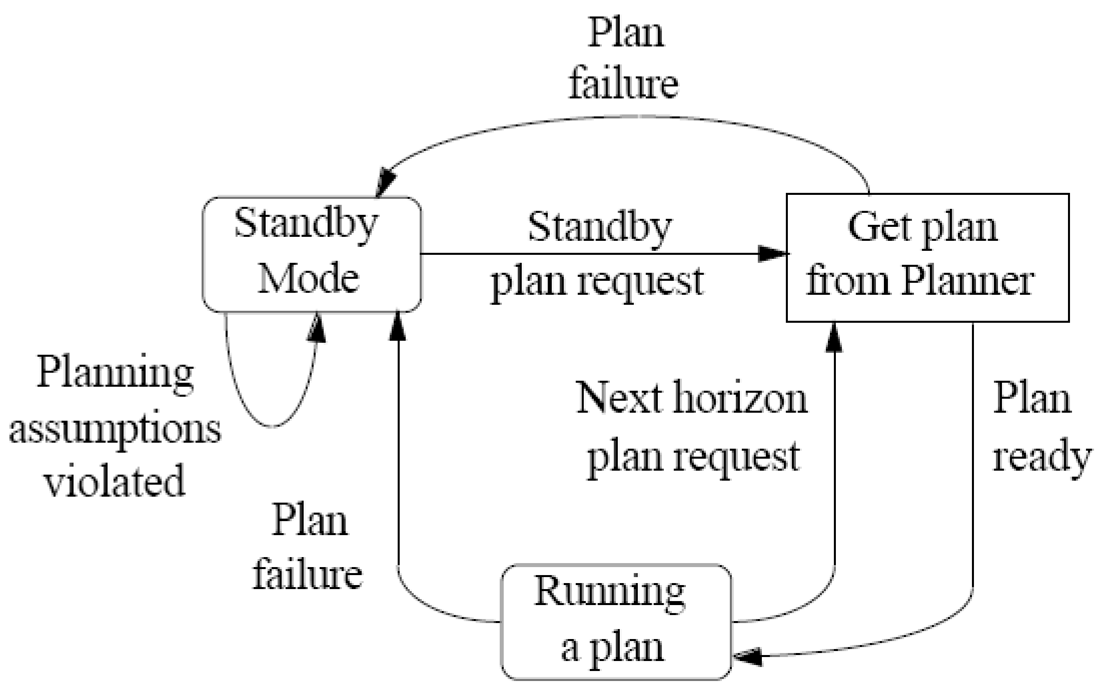
Remote Agent

- Planning to plan



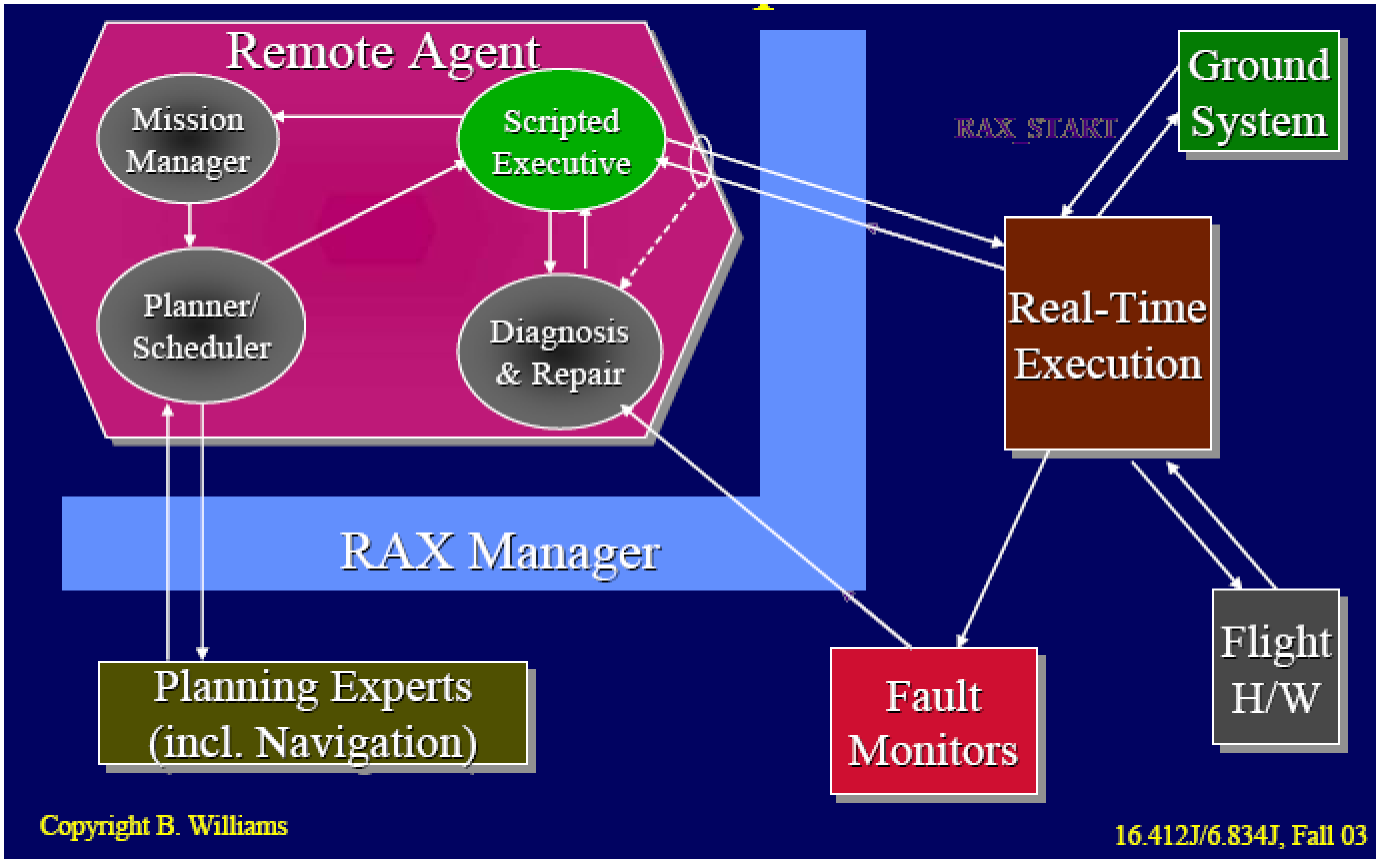
Remote Agent

- Periodic planning and replanning



Remote Agent

- Executive system dispatch tasks

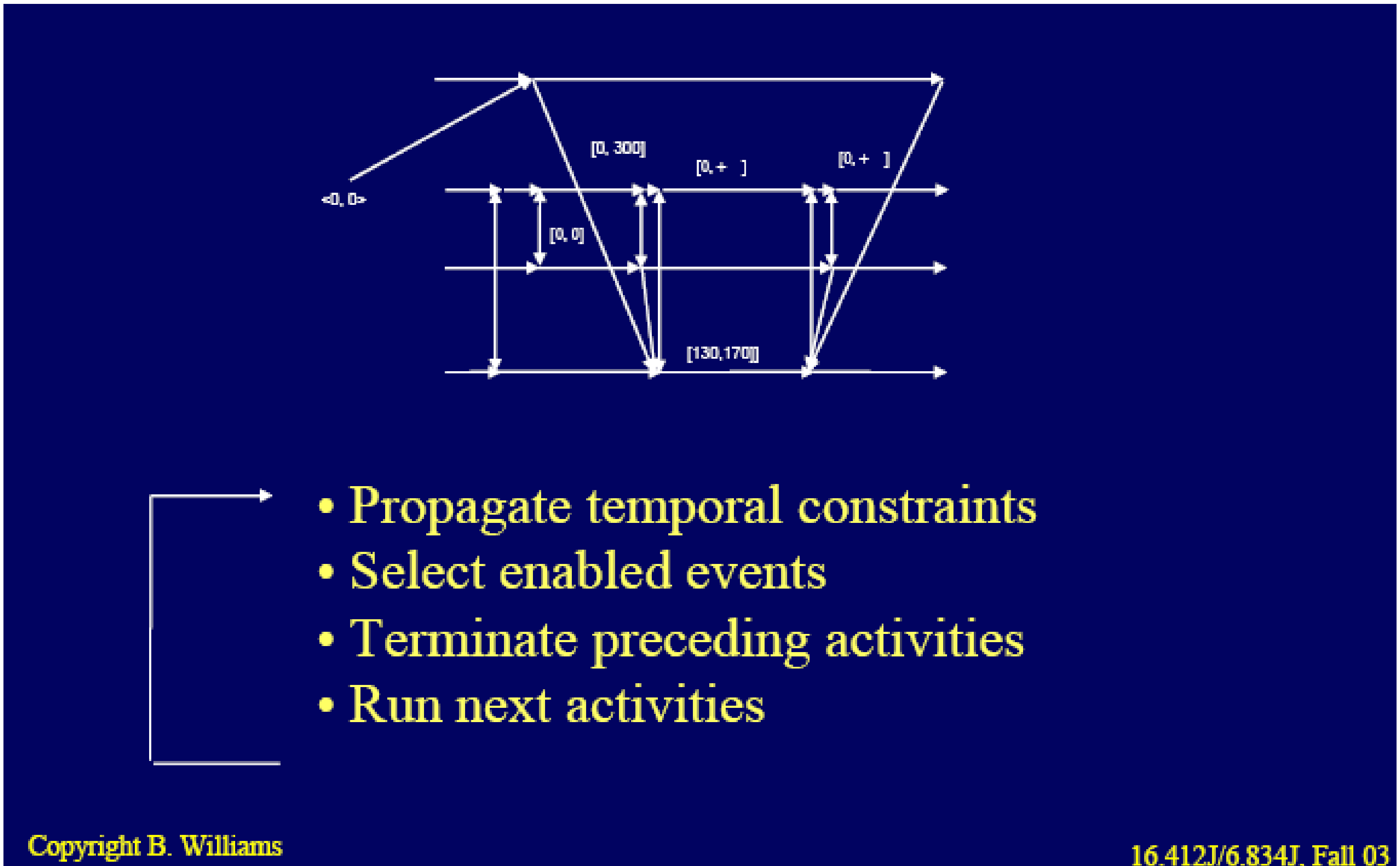


Remote Agent

- The Plan Executor has two duties:
  - Select and Schedule activities for execution
  - Update the network (constraint propagation) after the action execution or execution step (latency)
- Executor Cycle:
  - Activity Graph (STN) from Planner
  - Propagate with latency
  - Enabled time points = scheduled parents (fixed time points)
  - Select and Schedule enabled time points
  - Propagate constraint network given the new binds

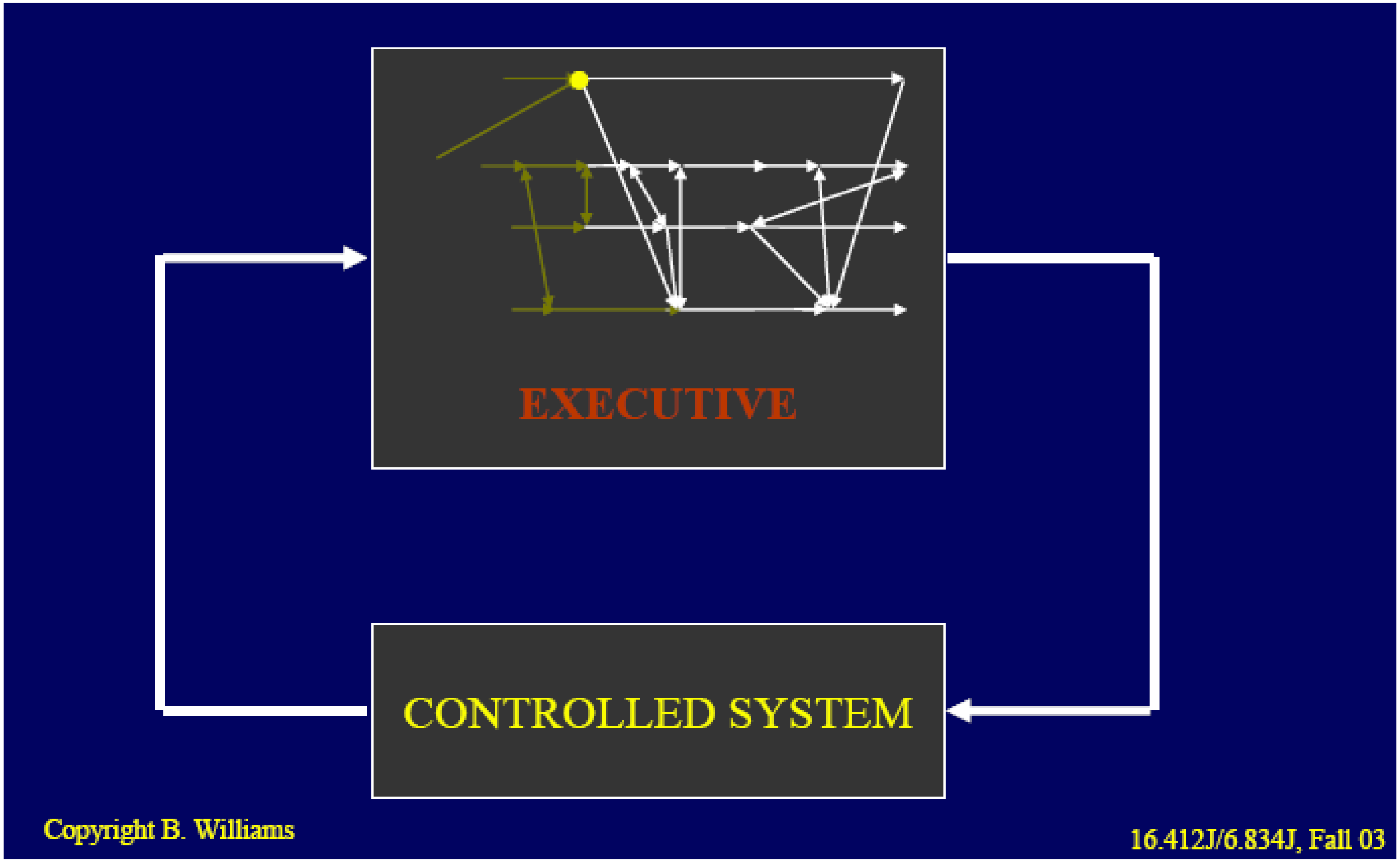
Remote Agent

- Executing Flexible Plans



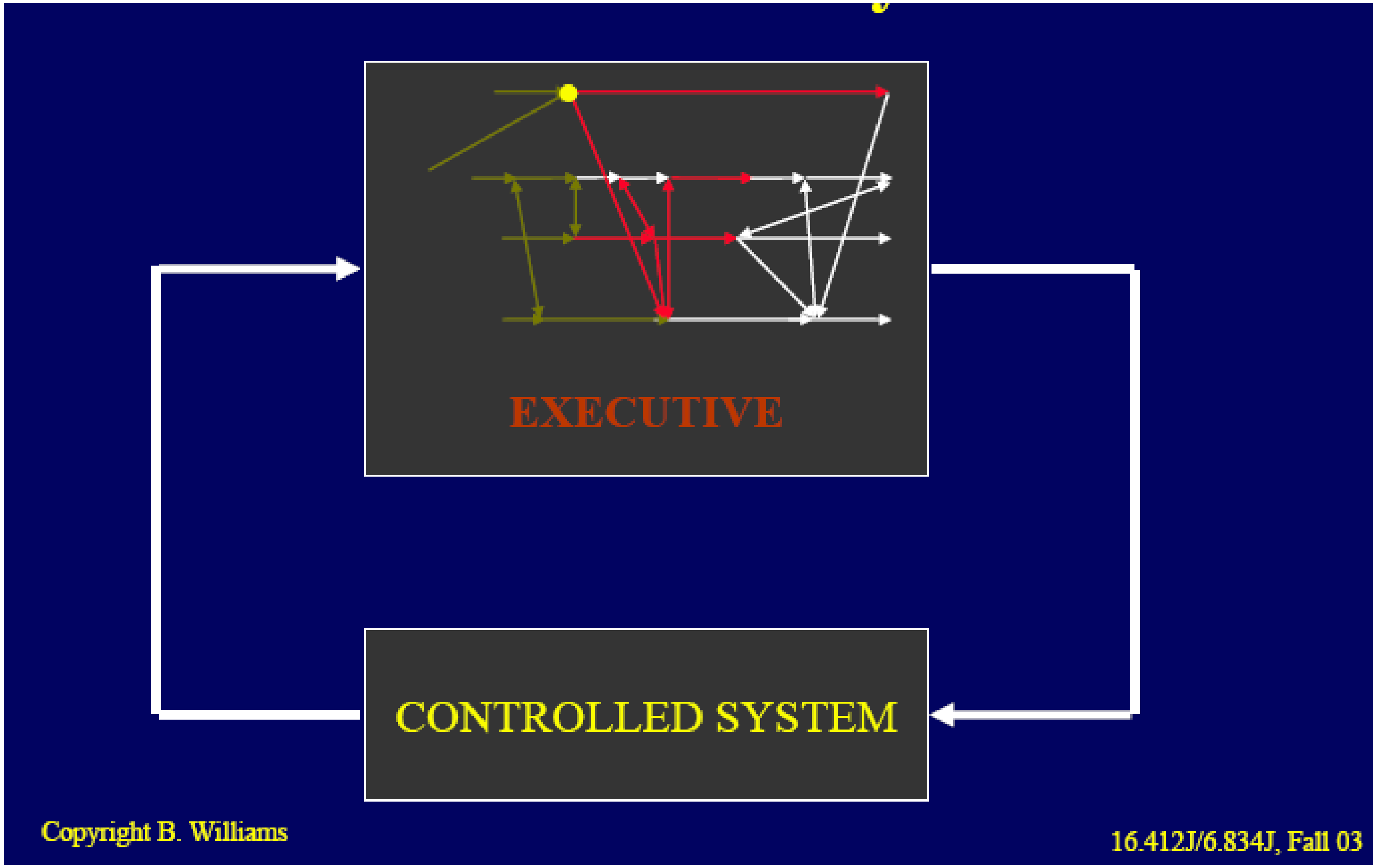
Remote Agent

- Constraint propagation can be costly



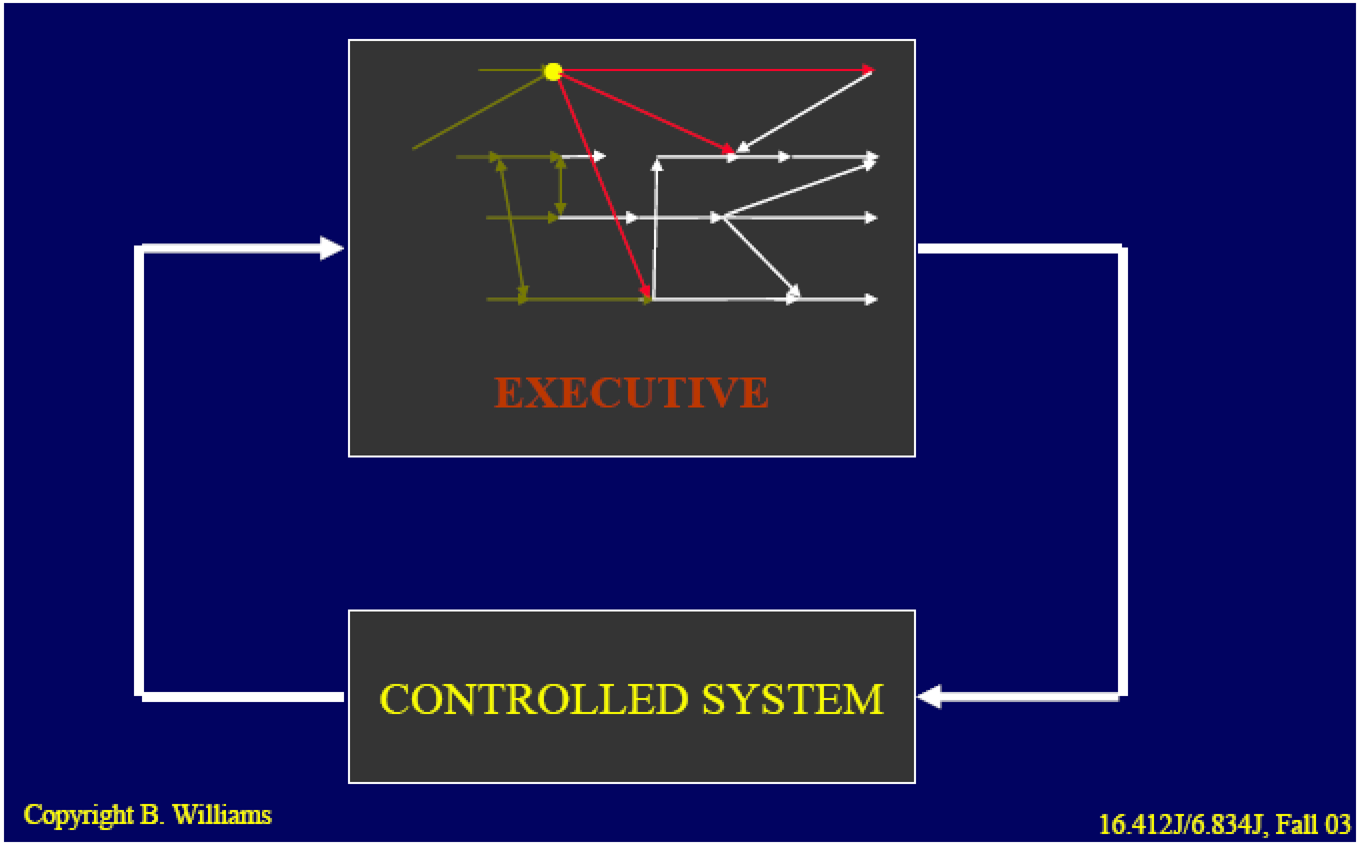
Remote Agent

- Constraint Propagation can be costly



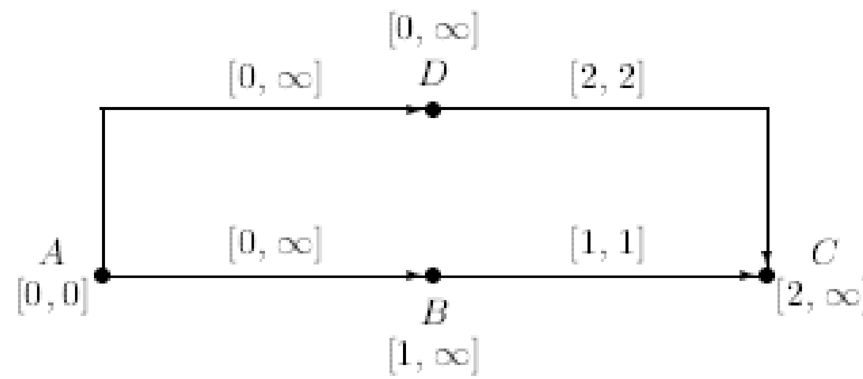
Remote Agent

- Solution: compile temporal constraints to an efficient network

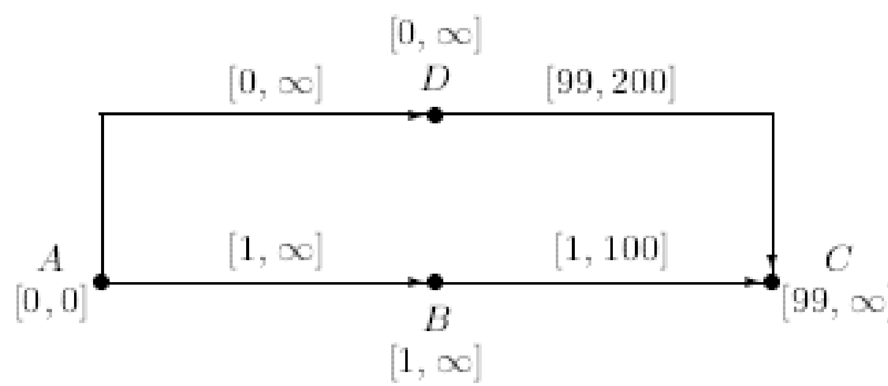


Remote Agent

- Dispatchability
  - Alcuni vincoli non visibili a tempo di esecuzione;

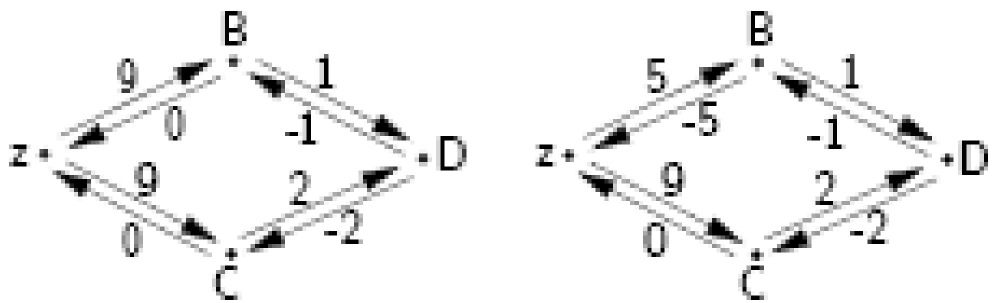


  - Occorre rendere la rete dispatchable aggiungendo vincoli impliciti (e.g. D prima di B)
  - Compilare la rete in forma dispatchable:
    - Introdotti vincoli impliciti
    - Tolti vincoli ridondanti



Dispatchability

A Sample Execution*



After executing B at time 5, it turns out that C must be executed at time 4 (which is already past).

* (Muscettola, Morris, & Tsamardinos 1998)

Dispatcher

Greedy Dispatcher*

While some time-points not yet executed:

- Wait until some time-point is executable.
- If more than one, pick one to execute.
- Propagate updates only to *neighboring* time-points (i.e., do not fully update $\mathcal{D}$).

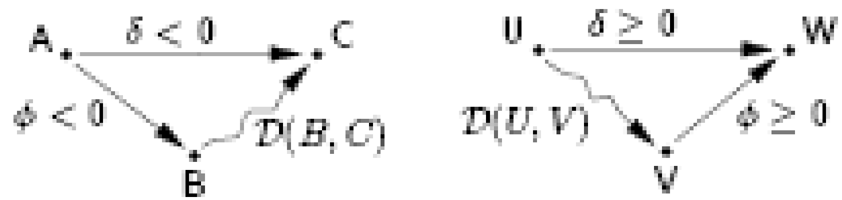
* (Muscettola, Morris, & Tsamardinos 1998)

Dispatcher

```
TIME DISPATCHING ALGORITHM:
    1. Let
         A = {start_time_point}
         current_time = 0
         S = {}
    2. Arbitrarily pick a time point TP in A such
       that current_time belongs to TP's time bound;
    3. Set TP's execution time to current_time and add
       TP to S;
    4. Propagate the time of execution
       to its IMMEDIATE NEIGHBORS in the distance
       graph;
    5. Put in A all time points TPx such that all
       negative edges starting from TPx have a
       destination that is already in S;
    6. Wait until current_time has advanced to
       some time between
          min{lower_bound(TP) : TP in A}
       and
          min{upper_bound(TP) : TP in A}
    7. Go to 2 until every time point is in S.
```

Dispatchability

Lower and Upper Dominance*



- The *negative edge* AC is *lower-dominated* if:
  $\delta = \phi + \mathcal{D}(B, C)$.
- The *non-negative edge* UW is *upper-domin'd* if:
  $\delta = \mathcal{D}(U, V) + \phi$.

* (Muscettola, Morris, & Tsamardinos 1998)

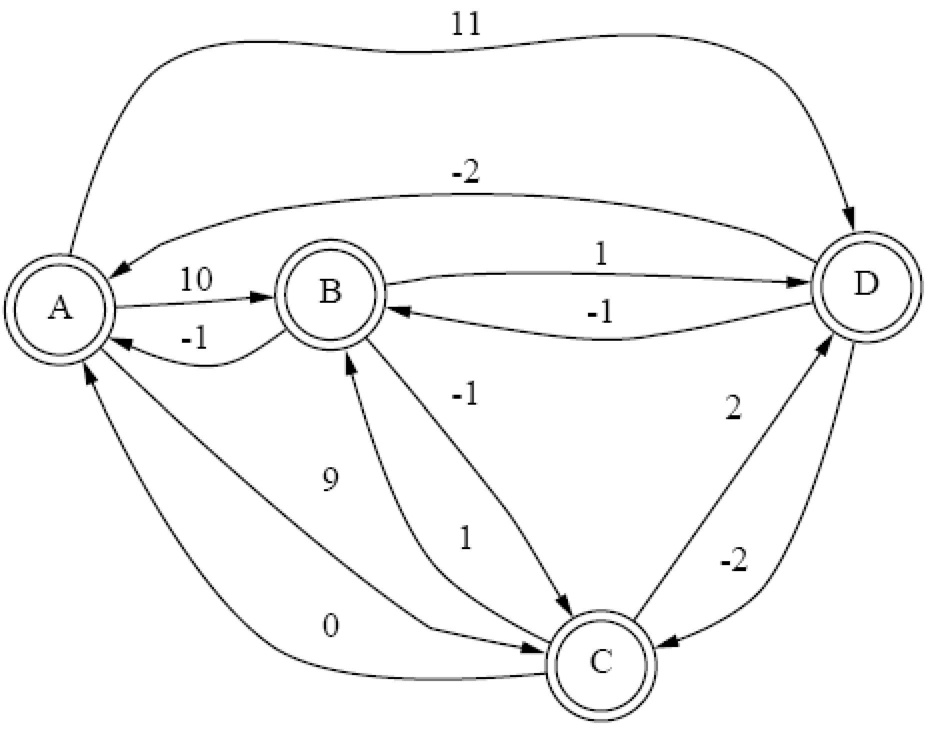
Dispatchability

Dispatchability*

- An STN that is guaranteed to be satisfied by the Greedy Dispatcher is called *dispatchable*.
- Any *consistent* STN can be transformed into an equivalent *dispatchable* STN.
- Step I: The corresponding All-Pairs graph is equivalent and dispatchable.
- Step II: Remove *lower- and upper-dominated* edges (does not affect dispatchability).

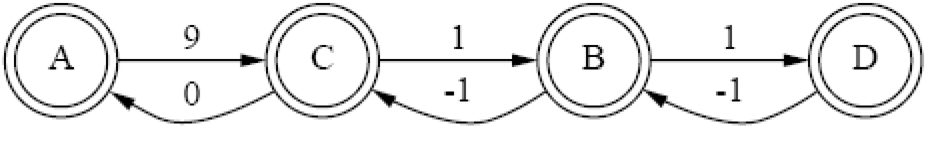
* (Muscettola, Morris, & Tsamardinos 1998).

Dispatchability



All pair graph

Filtered graph



Controllability

- Alcune attività non sono controllabili, ma solo osservabili
- E.g. after start_turn, end_turn ? Quando finisce?
- Il grafo delle attività STN contiene time point controllabili e non controllabili
- Le attività non controllabili non possono essere schedulate, ma solo osservate
- Propagazione??

Controllability

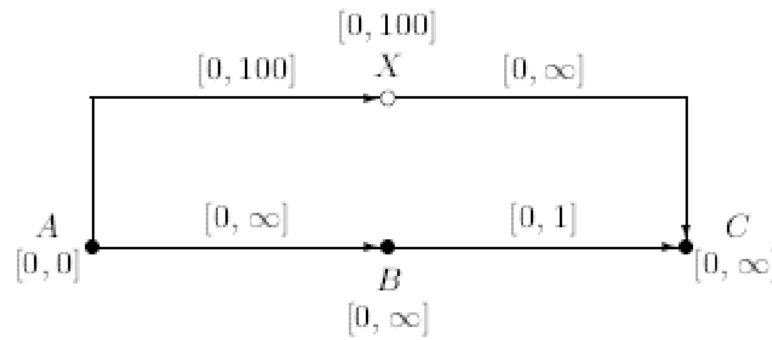
Controllability Issues*

- In real-world applications, an agent may only control some time-points directly; others may be controlled by other agents or Nature.
- Such a network is called *controllable* if there exists a strategy for the agent to execute the time-points under its direct control that will ensure the consistency of the network—no matter how the other agents or Nature execute their time-points.

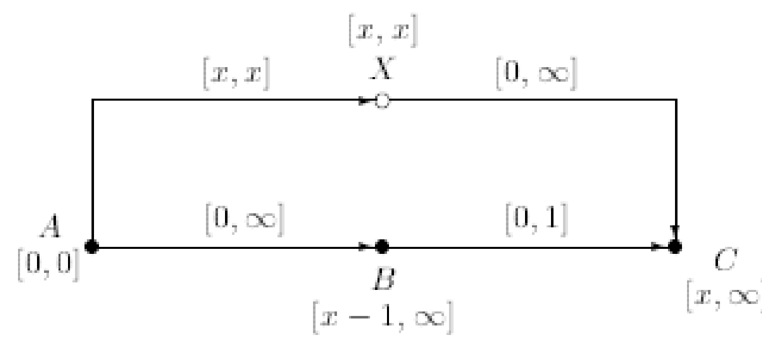
* (Vidal & Ghallab 1995; Vidal & Fargier)

Controllability

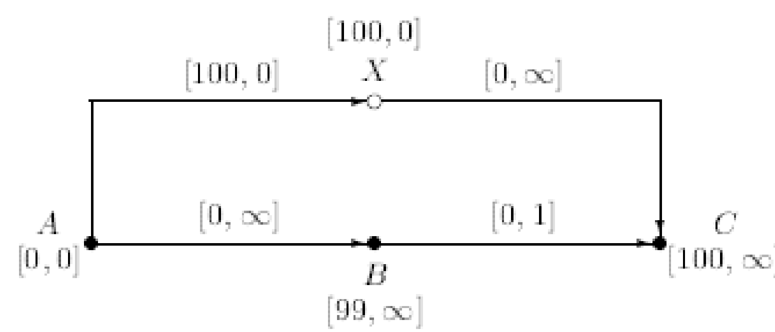
- Gestire eventi non controllabili
- Es. Se B schedulato prima di X, B vincola X



  – Soluzione Dinamica: B dopo X



  – Soluzione Forte: B a 99



Controllability

- **Weak Controllability**: per ogni evento incontrollabile esiste uno scheduling che permette l'esecuzione;
- **Strong Controllability**: esiste uno scheduling robusto qualunque siano gli eventi non controllabili;
- **Dynamic Controllability**: per ogni evento incontrollabile passato esiste uno scheduling che permette l'esecuzione.

Controllability

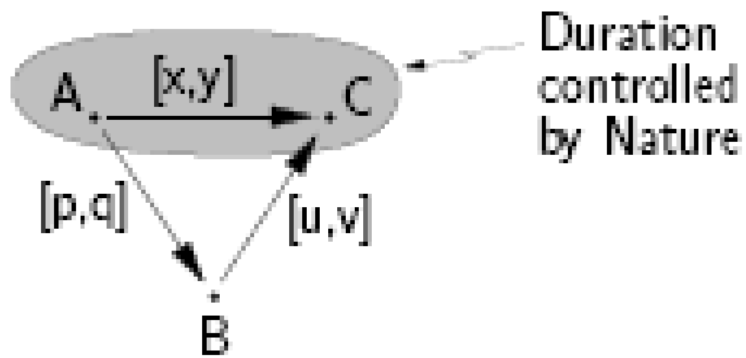
Checking Dynamic Controllability*

Morris et al. (2001) present a sound and complete algorithm for checking dynamic controllability using:

- *Triangular Reductions*
- *Wait Propagation*

Controllability

Triangular Reductions



- If $v < 0$, then B follows C – no reduction.
- If $u \geq 0$, then B precedes C. Must tighten bounds on interval AB to: [y-v, x-u].

Controllability

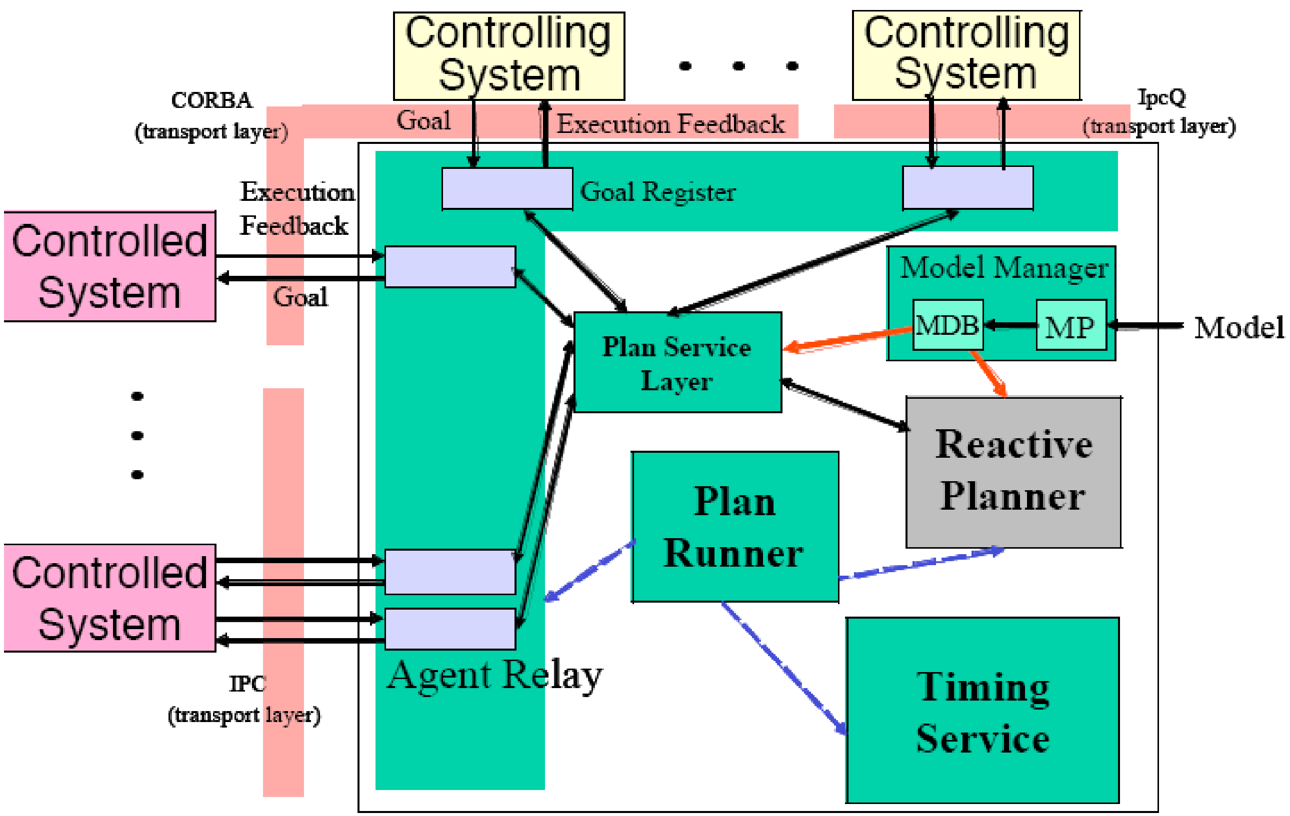
Triangular Reductions (ctd.)

- If $u < 0$ and $v \geq 0$, then the order of B and C is not yet determined. Derive a *WAIT*: If C has not yet been executed, B must *wait* to be executed until (y-v) after A.

Waits can be propagated much like binary constraints.

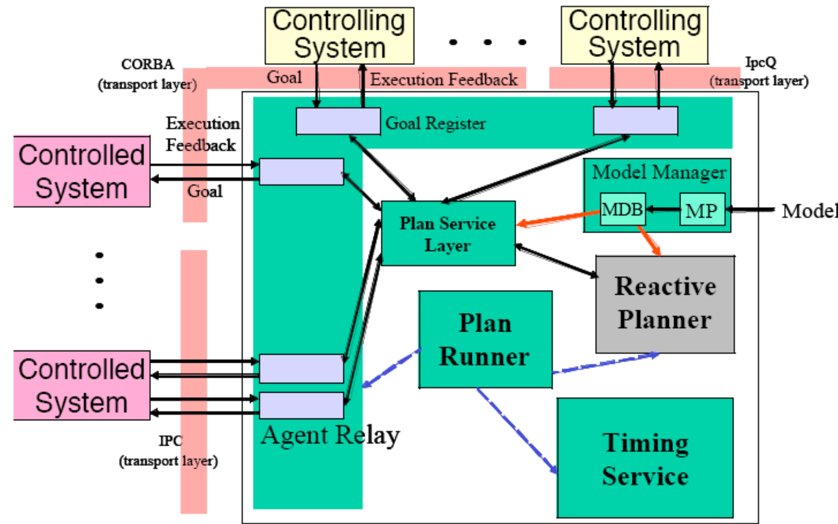
IDEA Architecture

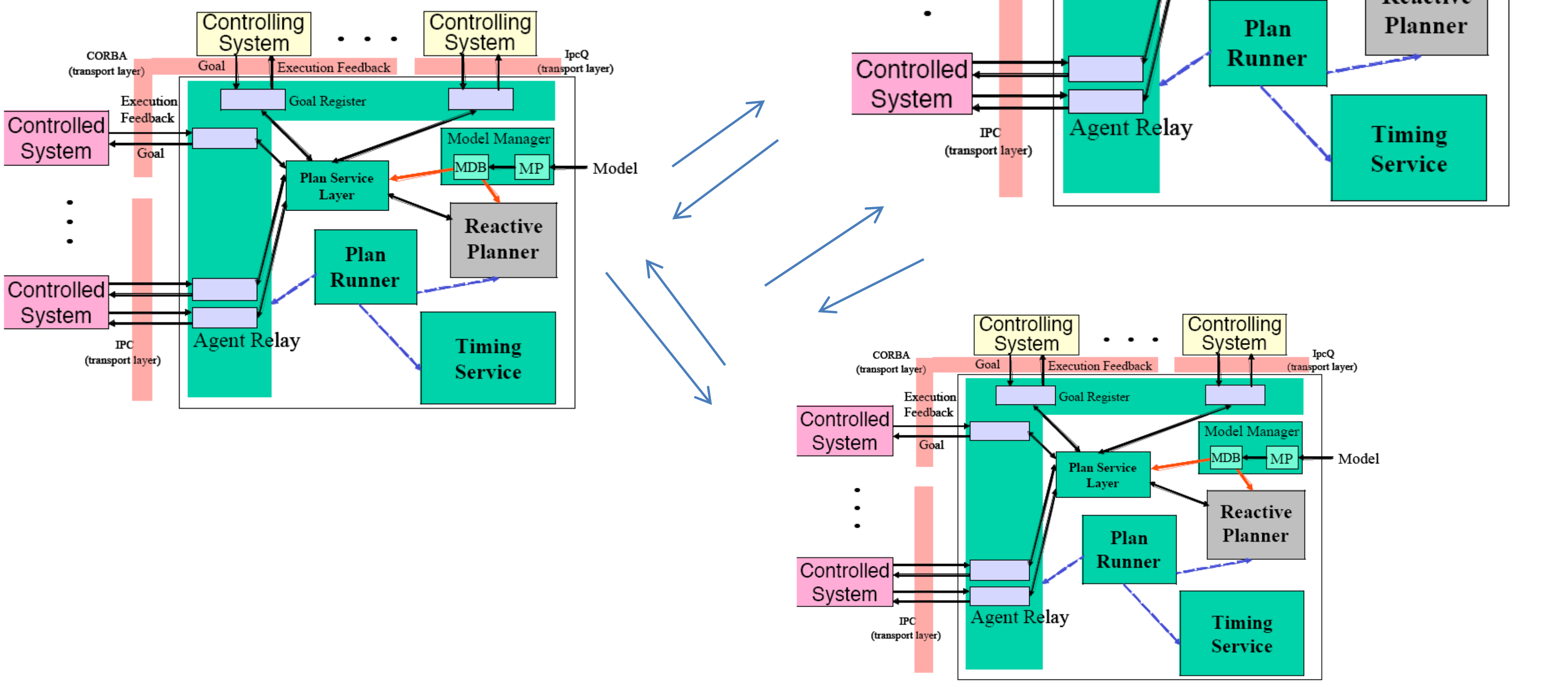
- Evoluzione del RA: reactive and deliberative planning



IDEA Architecture

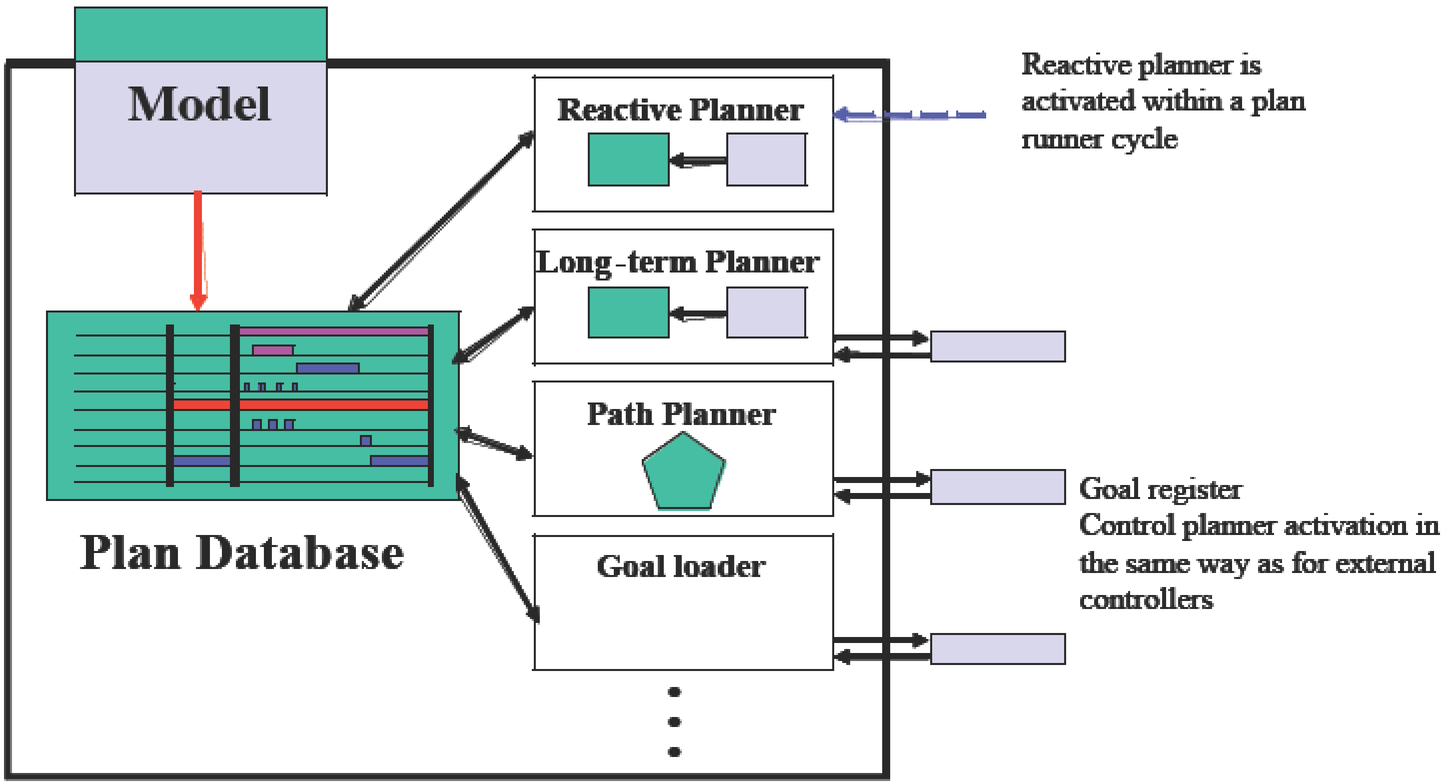
- Muti-agent architecture:





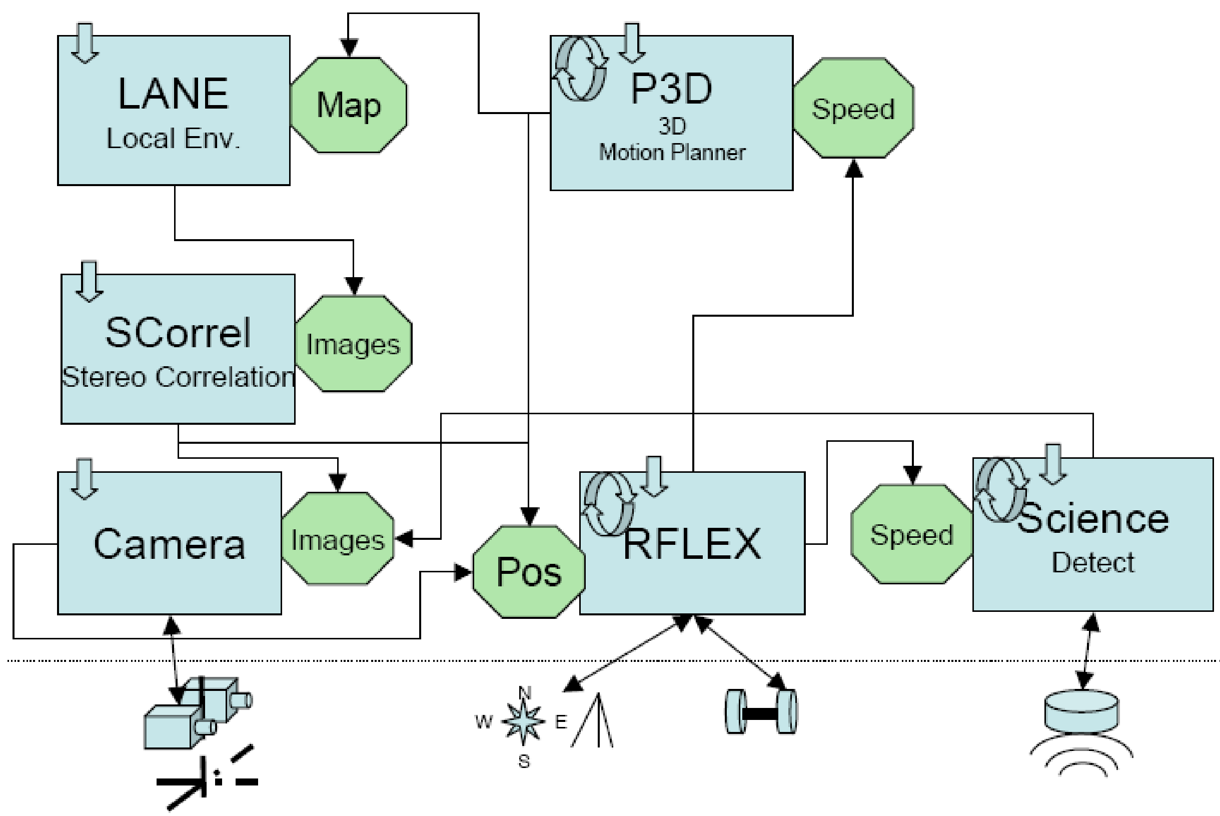
IDEA Architecture

- Reactive planning come controllo
- Interazione deliberative and reactive planning



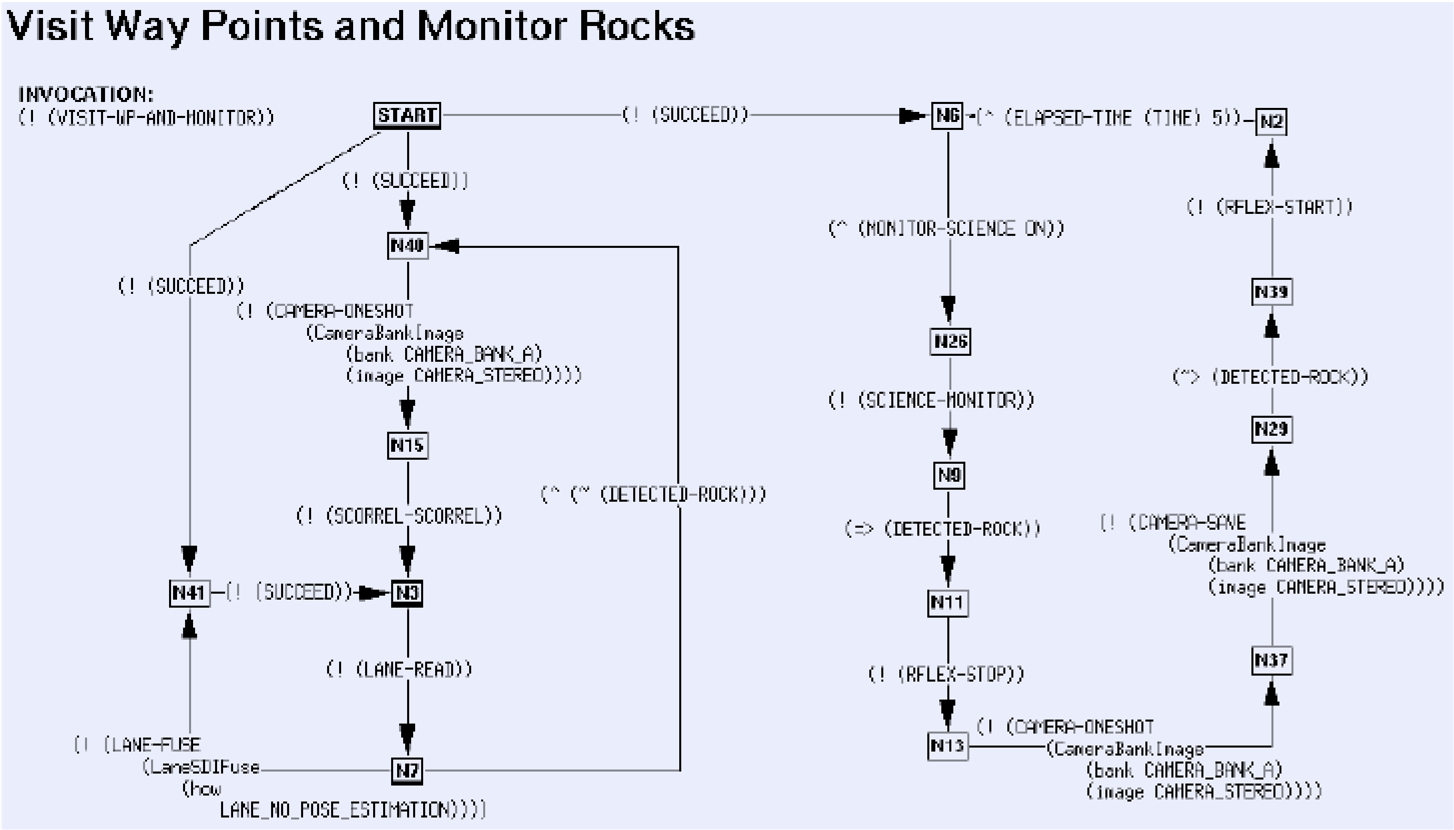
Functional Layer

- GenoM (LAAS)



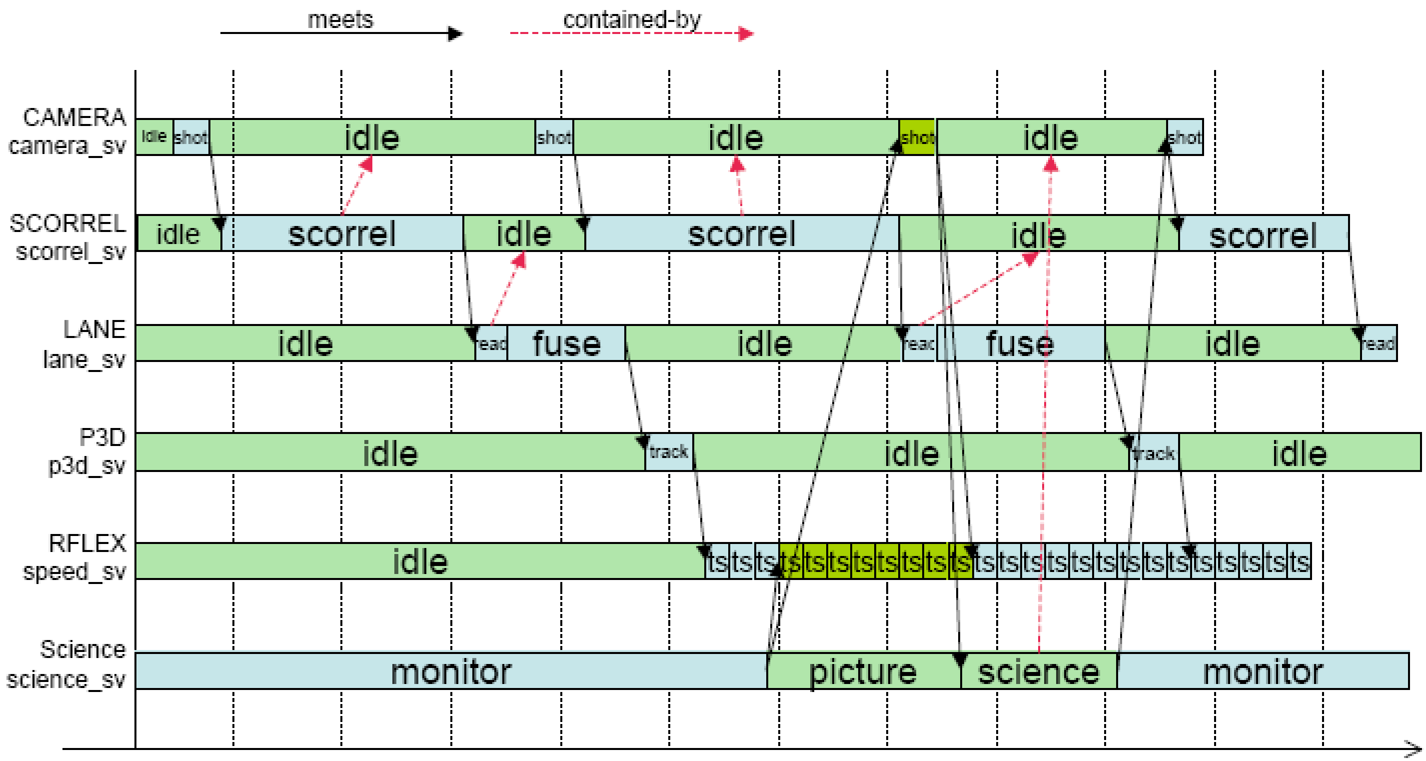
PRS Controller

- A procedural controller (vedi dopo ...)



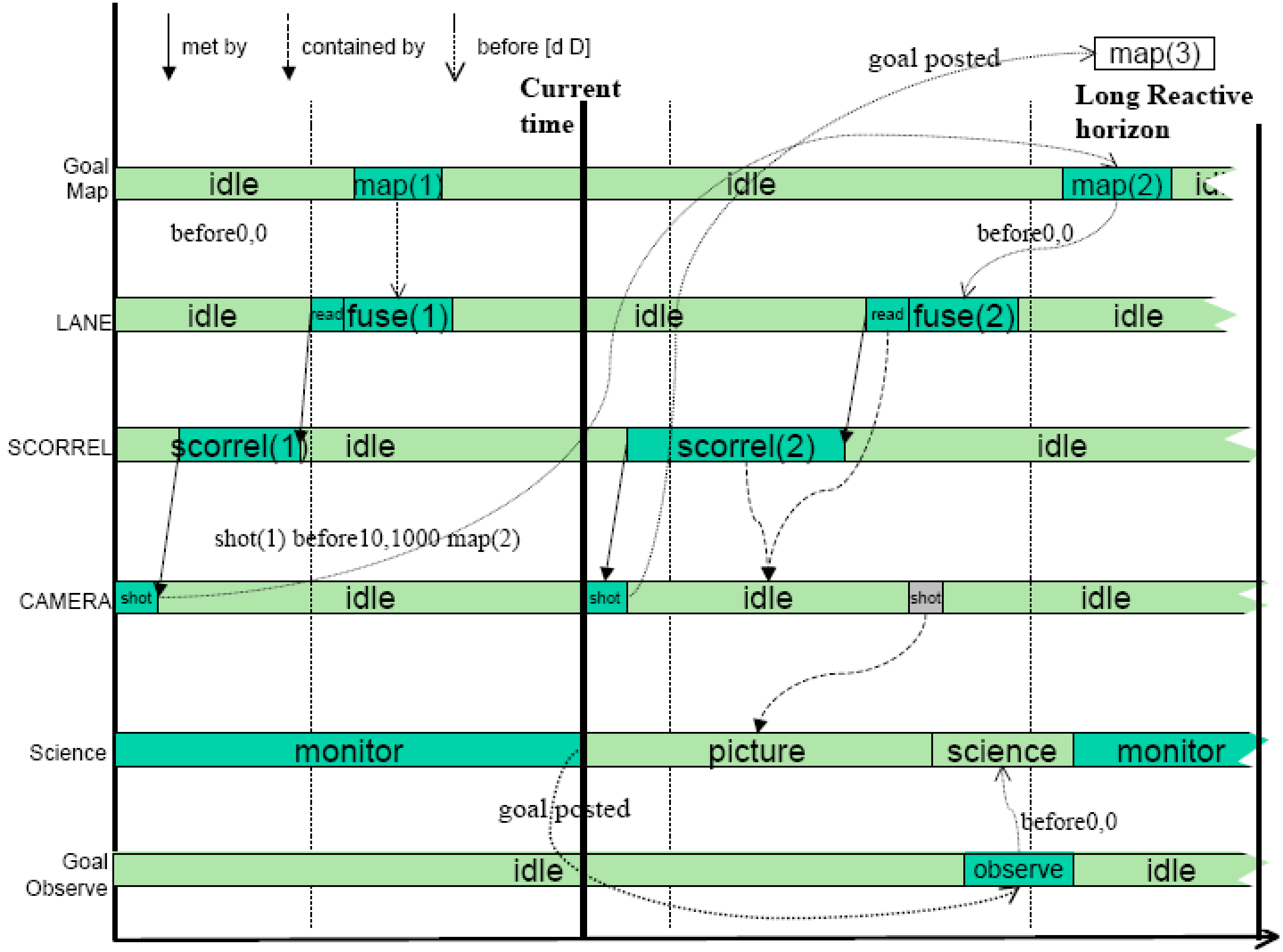
IDEA Architecture

- Attività pianificate (plan database):



IDEA Architecture

- Reactive Planning



IDEA Architecture

- Reactive and Deliberative planning