Temporal Planning

Task Planning

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

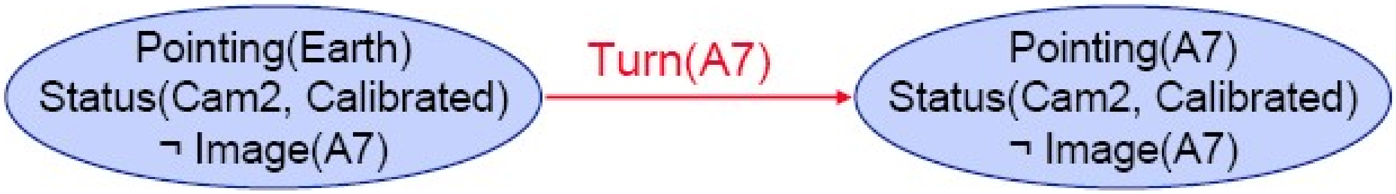
Extensions

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

Model

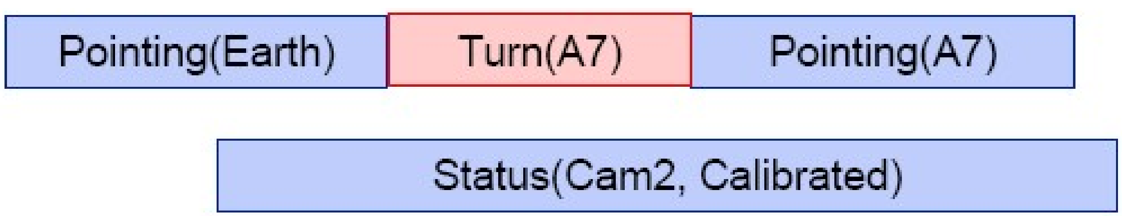
State-centric (Mc Carthy):

for each time describe propositions that are true



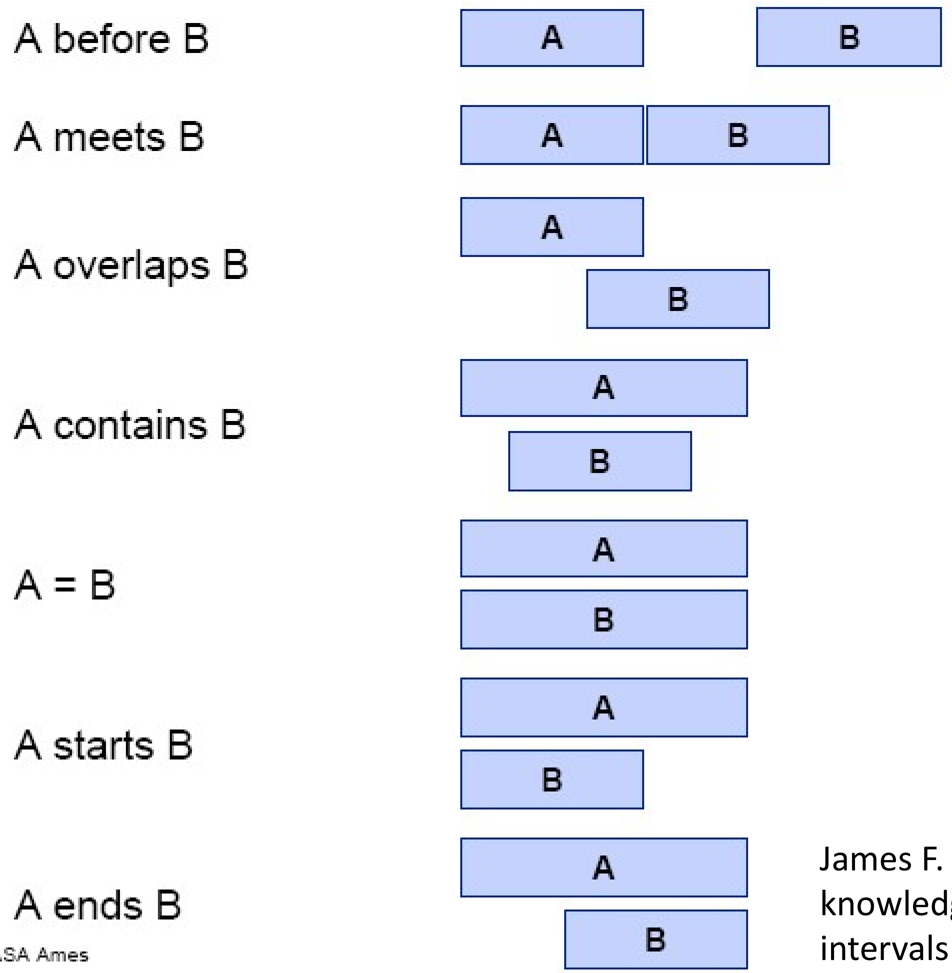
History-based (Hayes):

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

James F. Allen, "Maintaining knowledge about temporal intervals 1983

Time Points vs Intervals

- Sentences:
  - We found the letter at twelve noon
  - We found the letter yesterday
  - We found the letter while John was away
  - We found the letter after we made the decision

Interval Relation	Equivalent Relations on Endpoints
t < s	t+ < s-
t = s	(t- = s-) & (t+ = s+)
t overlaps s	(t- < s-) & (t+ > s-) & (t+ < s+)
t meets s	t+ = s-
t during s	((t- > s-) & (t+ =< s+)) or ((t- >= s-) & (t+ < s+))

James F. Allen, "Maintaining knowledge about temporal intervals 1983

Relation	Symbol	Symbol for Inverse	Pictoral Example
X *before* Y	<	>	XXX YYY
X *equal* Y	=	=	XXX YYY
X *meets* Y	m	mi	XXXYYY
X *overlaps* Y	o	oi	XXX YYY
X *during* Y	d	di	XXX YYYYYY
X *starts* Y	s	si	XXX YYYYY
X *finishes* Y	f	fi	XXX YYYYY

Temporal Operators

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

↓

TakeImage (?target, ?instr)

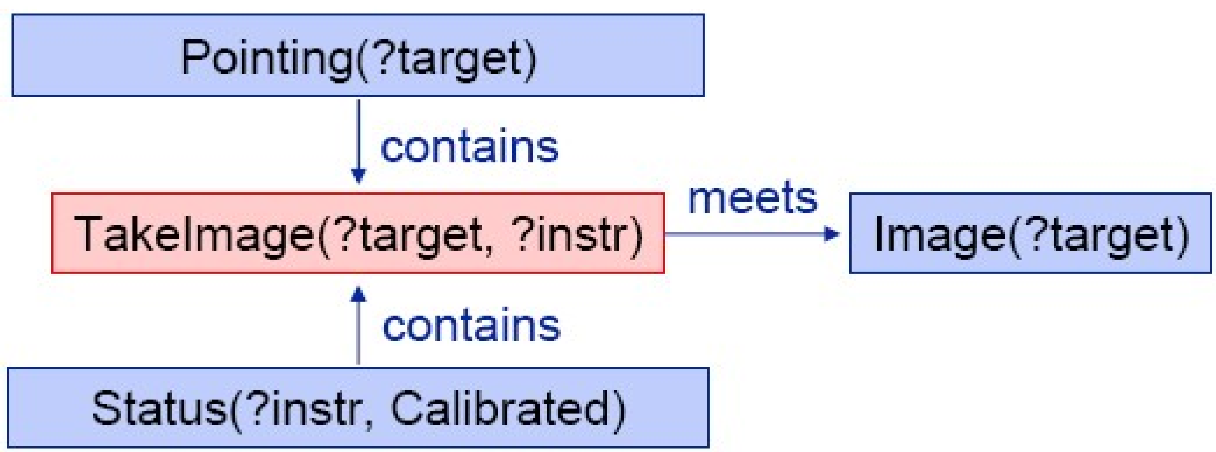
contained-by	Status(?instr, Calibrated)
contained-by	Pointing(?target)
meets	Image(?target)

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)



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)

⬇

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

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

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

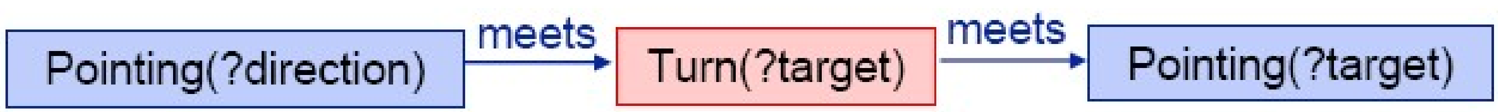
$\wedge\ \exists R\ \{\text{Image}(?\text{target})_R \ \wedge\ \text{Meets}(A, R)\}$

Based on slides by Dave Smith, NASA Ames

Temporal Operators

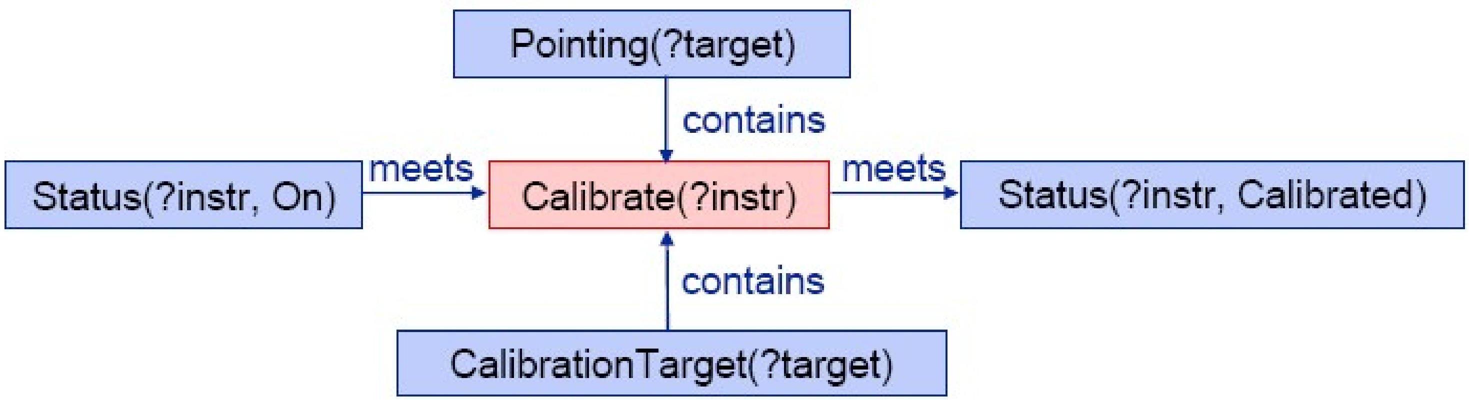
Turn (?target)

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



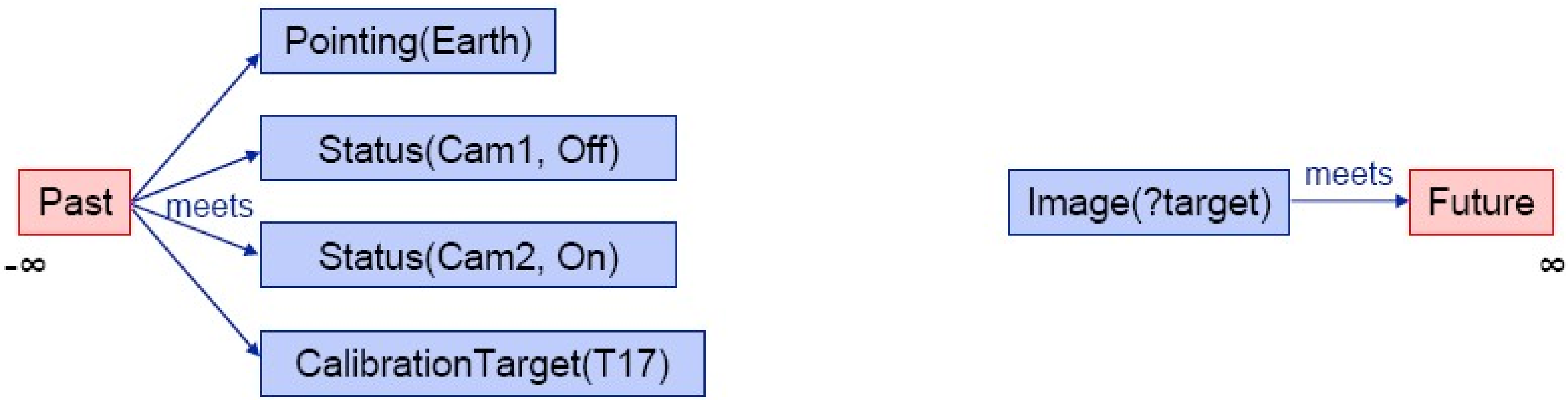
Temporal Operators

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



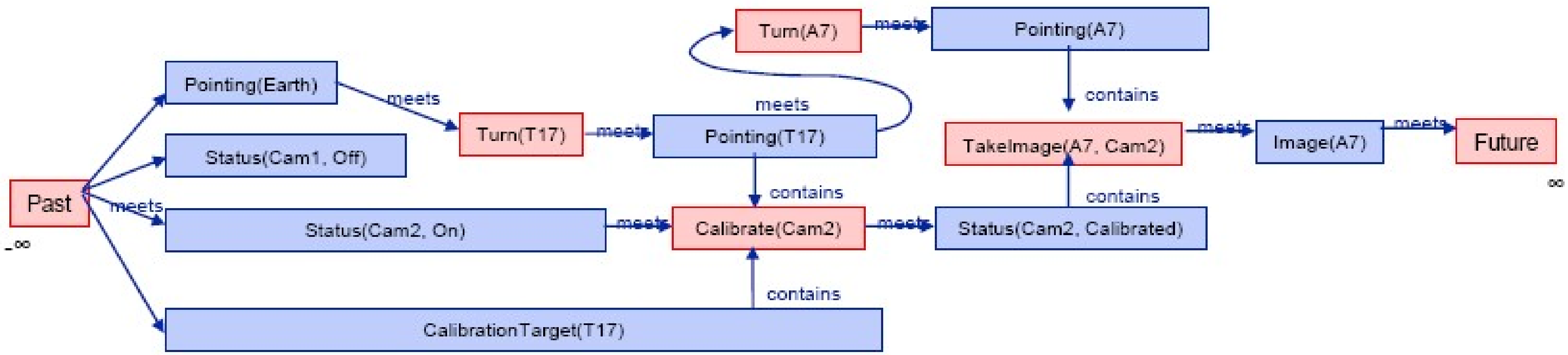
Based on slides by Dave Smith, NASA Ames

Temporal Planning Problem



Based on slides by Dave Smith, NASA Ames

Consistent Complete Plan



Based on slides by Dave Smith, NASA Ames

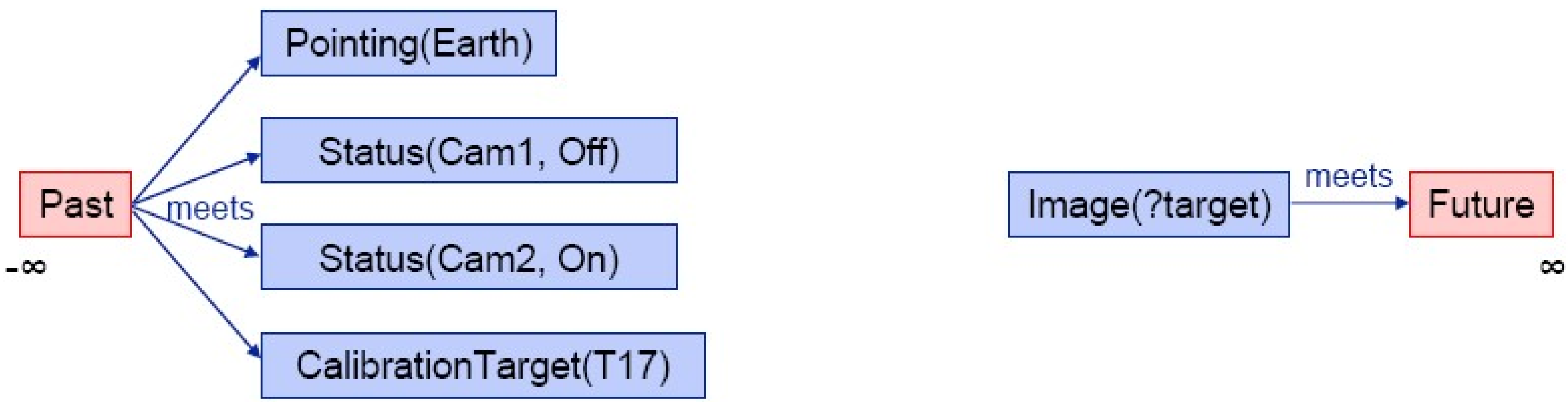
CBI-Planning

Choose:

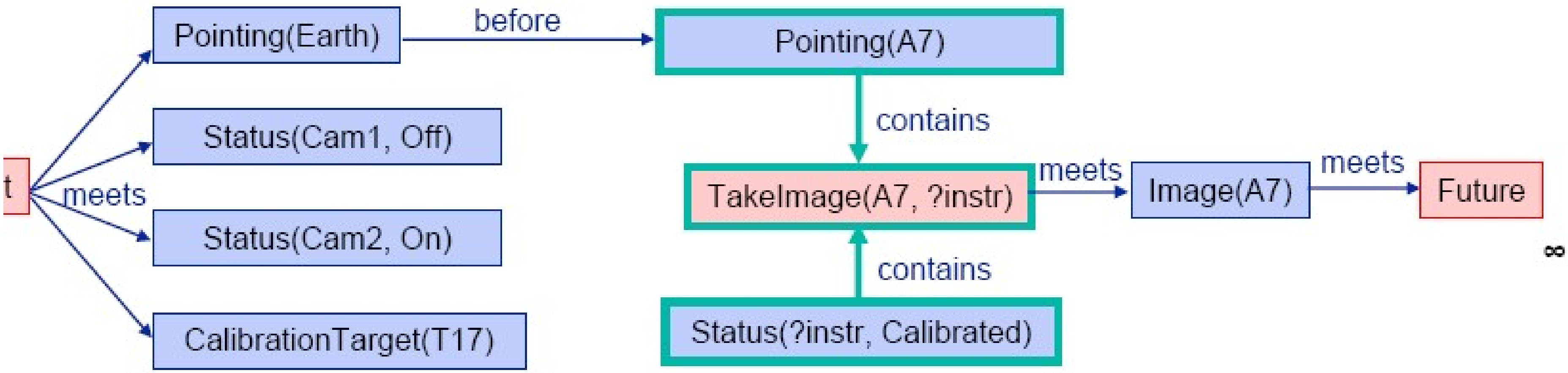
- introduce an action & instantiate constraints
- coalesce propositions

Propagate constraints

Initial Plan

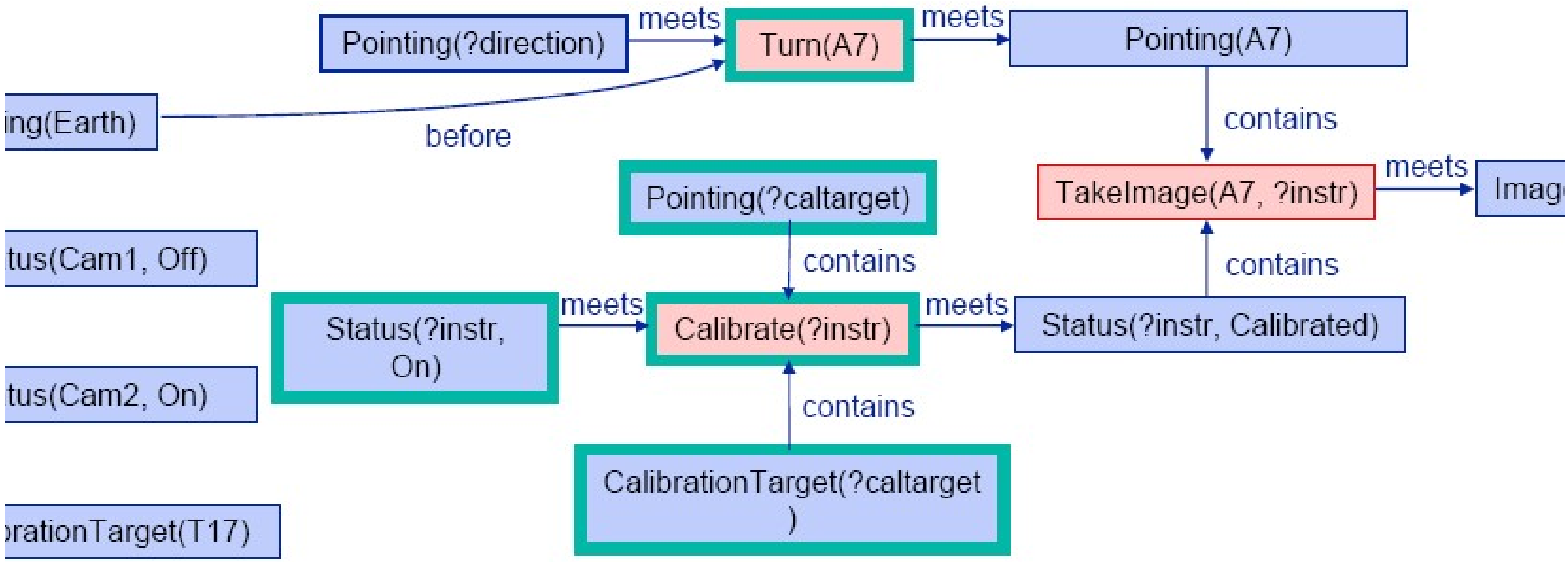


Expansion



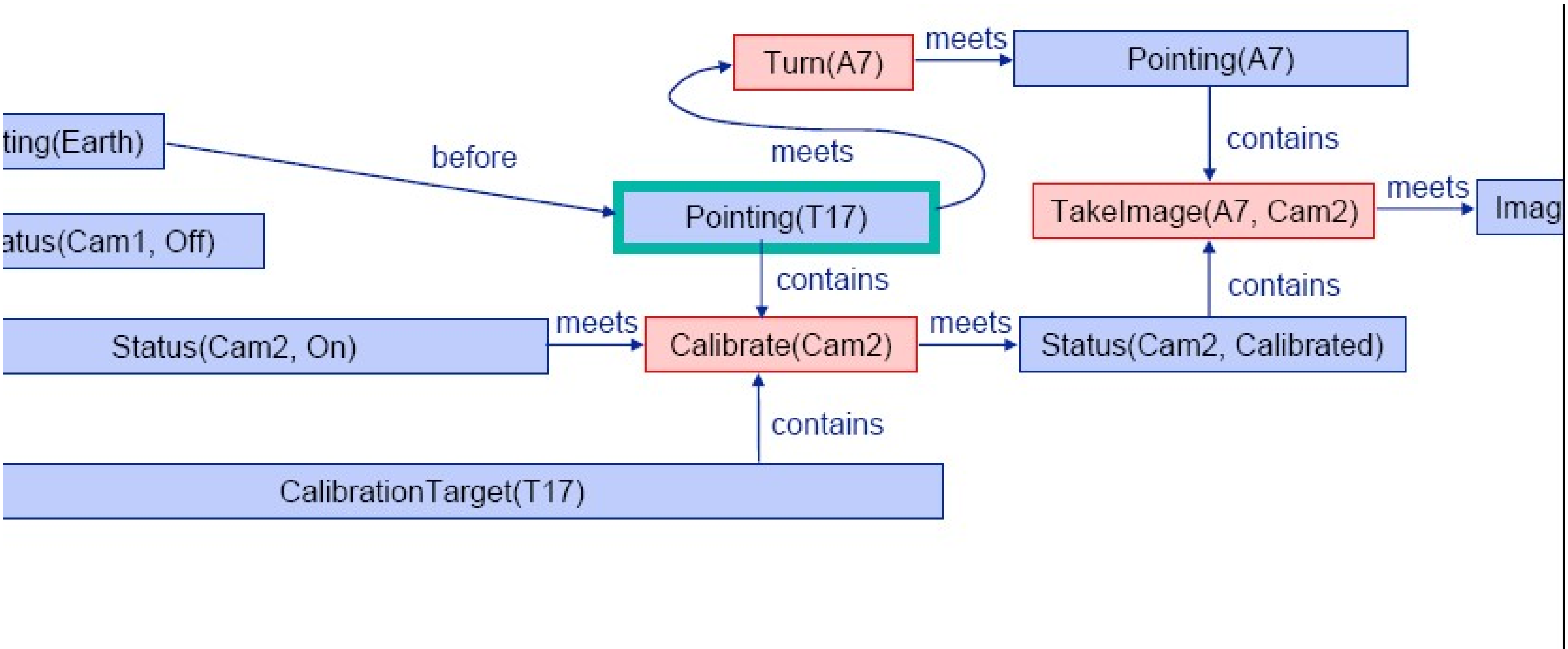
t
meets
Pointing(Earth)
before
Pointing(A7)
Status(Cam1, Off)
contains
Status(Cam2, On)
TakeImage(A7, ?instr)
meets
Image(A7)
meets
Future
∞
contains
CalibrationTarget(T17)
Status(?instr, Calibrated)

Expansion

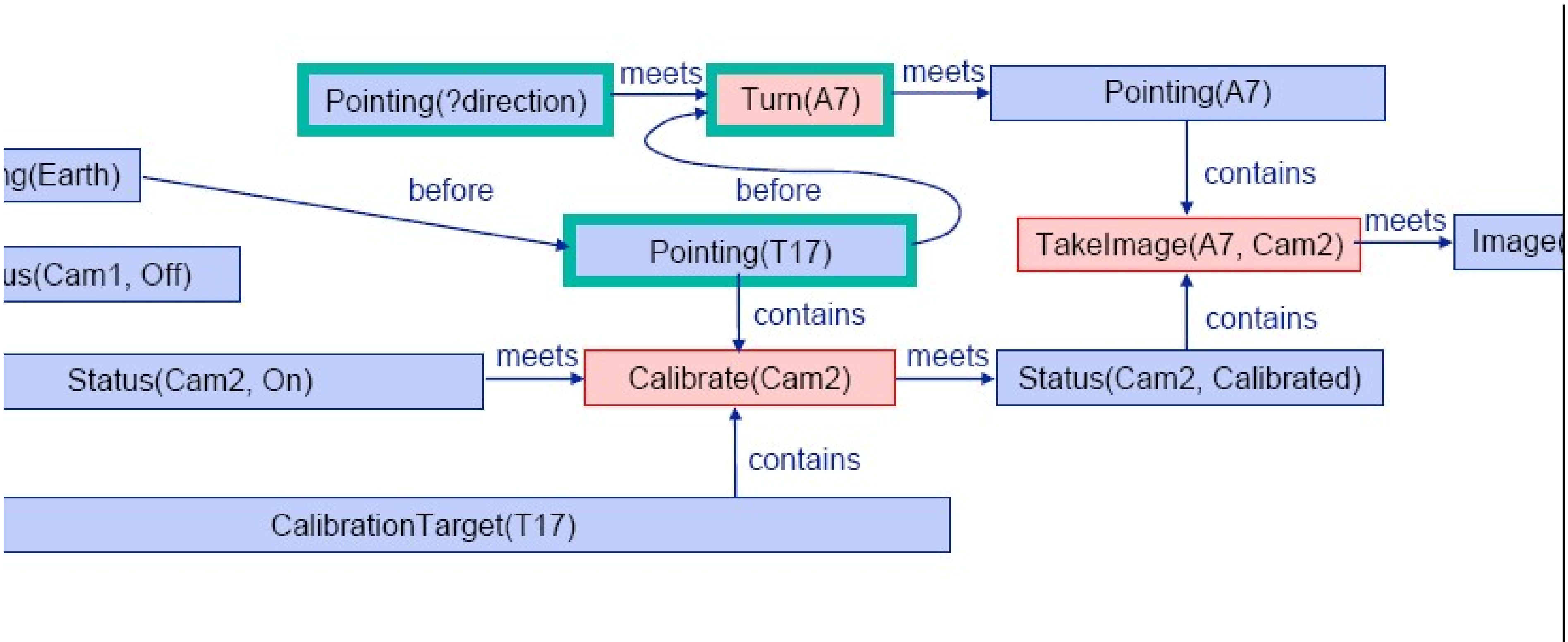


Based on slides by Dave Smith, NASA Ames

Coalescing

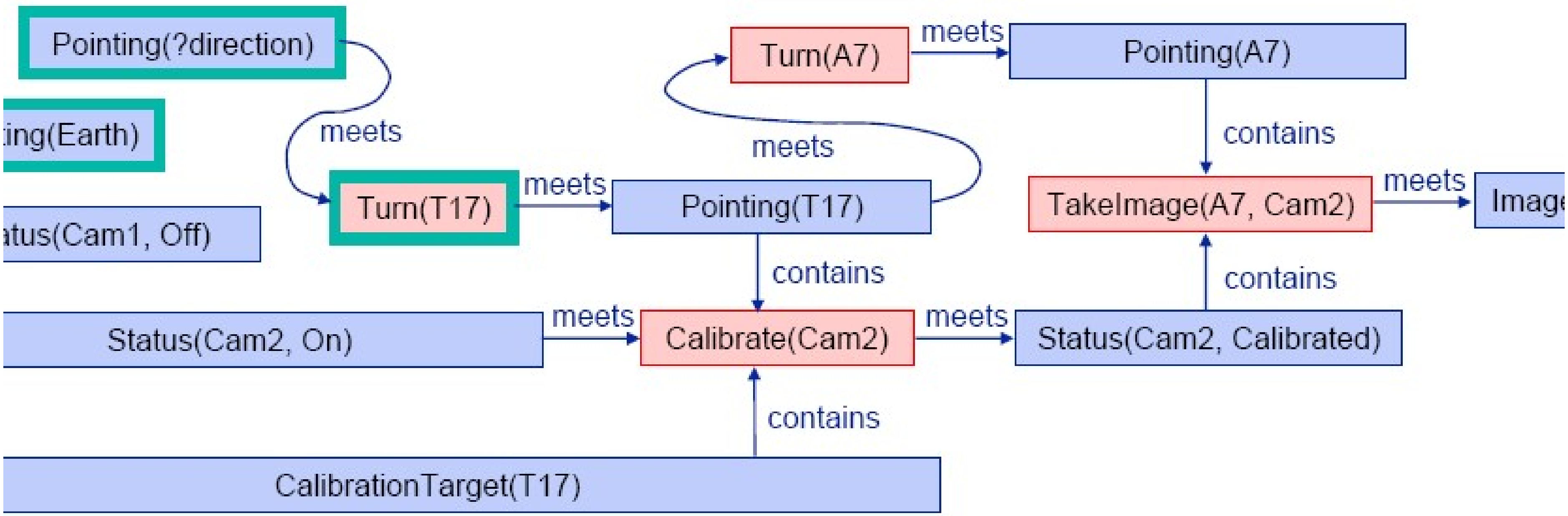

Turn(A7)
meets
Pointing(A7)
ting(Earth)
before
meets
contains
Pointing(T17)
TakeImage(A7, Cam2)
meets
Imag
atus(Cam1, Off)
contains
contains
Status(Cam2, On)
meets
Calibrate(Cam2)
meets
Status(Cam2, Calibrated)
contains
CalibrationTarget(T17)

Coalescing



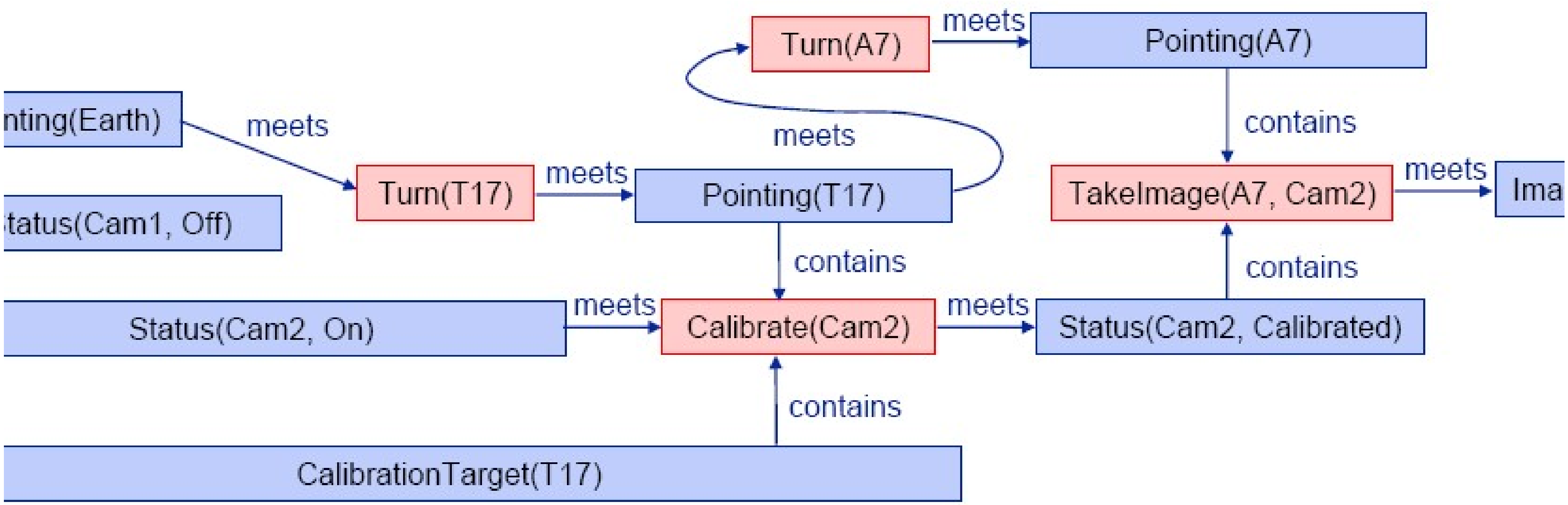
Pointing(?direction)
meets
Turn(A7)
meets
Pointing(A7)
ng(Earth)
before
before
contains
us(Cam1, Off)
Pointing(T17)
TakeImage(A7, Cam2)
meets
Image(
contains
contains
Status(Cam2, On)
meets
Calibrate(Cam2)
meets
Status(Cam2, Calibrated)
contains
CalibrationTarget(T17)

Expansion



Pointing(?direction)
ing(Earth)
atus(Cam1, Off)
meets
Turn(T17)
meets
Pointing(T17)
meets
Turn(A7)
meets
Pointing(A7)
contains
TakeImage(A7, Cam2)
meets
Image
contains
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)

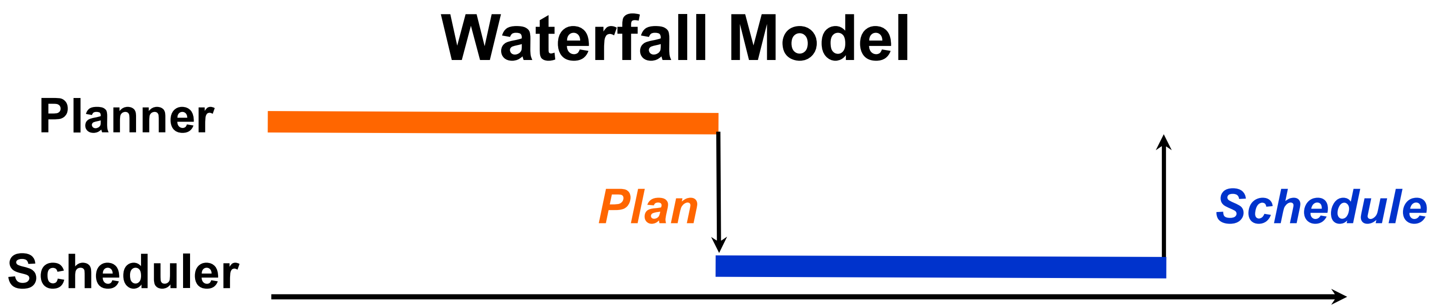
CBI-Planners

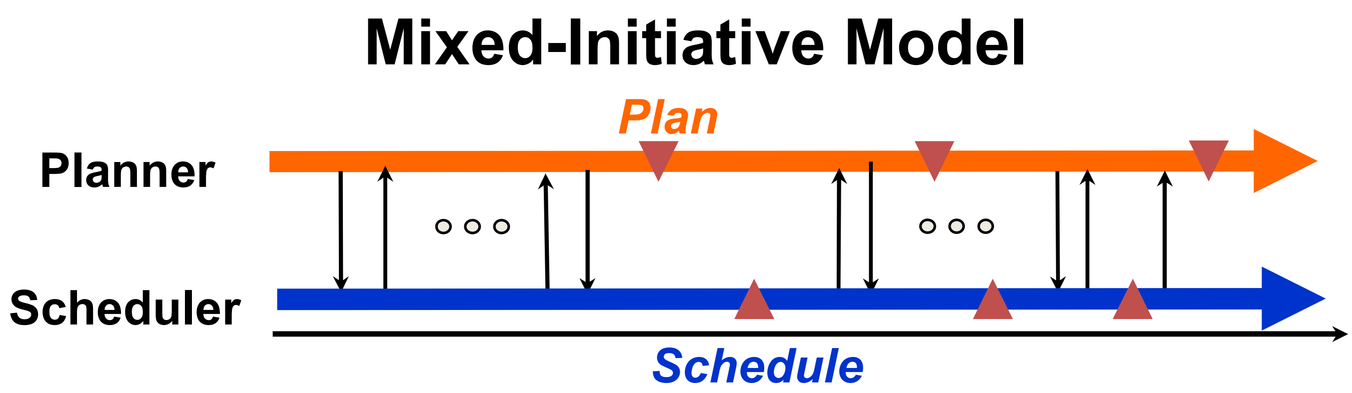
Zeno (Penberthy)	intervals, no CSP
Trains (Allen)	
Descartes (Joslin)	extreme least commitment
IxTeT (Ghallab)	functional rep.
HSTS (Muscettola)	functional rep., activities
EUROPA (Jonsson)	functional rep., activities

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

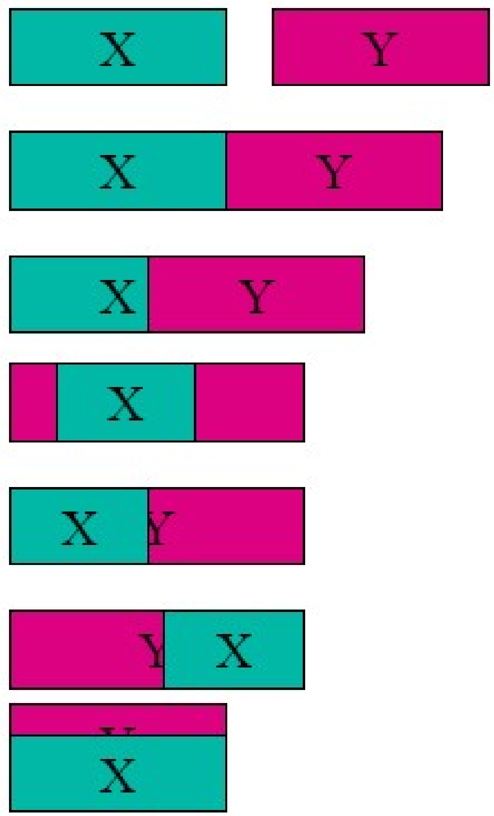
Planning & Scheduling





Temporal Constraints

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

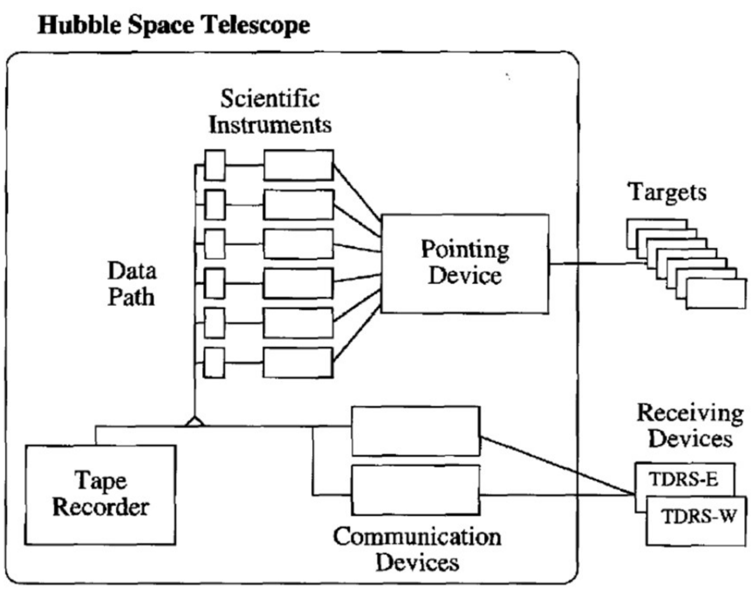


- y after x
- y met-by x
- y overlapped-by x
- y contains x
- y started-by x
- y finished-by x
- y equals x

HSTS: *Integrating* Planning and Scheduling

HSTS: Integrating Planning and Scheduling, Nicola Muscettola CMU-RI-TR-93-05

- HSTS (Heuristic Scheduling Testbed System):
  - representation and problem solving framework that provides an integrated view of planning and scheduling.
  - Leaving as much temporal flexibility as possible during the planning and scheduling process
  - planner and scheduler for short term scheduling of the Hubble Space Telescope



The pointing subsystem is responsible for orienting HST toward a target and locking it at the center of the field of view of the instrument.

6 scientific instruments, but power does not allow all of them to be operational simultaneously. Moving an instrument between operation and quiescence requires complex reconfiguration sequences

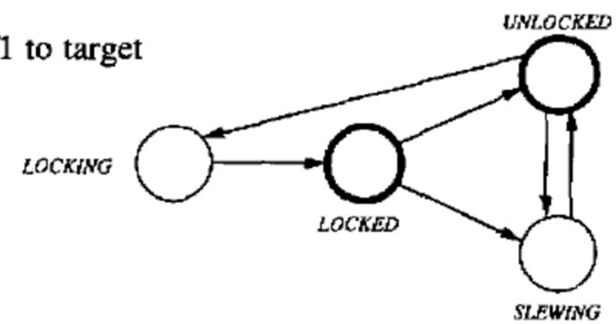
HSTS: *Integrating* Planning and Scheduling

HSTS: Integrating Planning and Scheduling, Nicola Muscettola CMU-RI-TR-93-05

- HSTS (Heuristic Scheduling Testbed System):
  - representation and problem solving framework that provides an integrated view of planning and scheduling.
  - Leaving as much temporal flexibility as possible during the planning and scheduling process
  - planner and scheduler for short term scheduling of the Hubble Space Telescope

- *UNLOCKED*(?*T*) : the telescope is pointing in the generic direction of target ?*T*;
- *LOCKED*(?*T*) : the telescope is actively tracking target ?*T*;
- *LOCKING*(?*T*) : the tracking device is locking onto target ?*T*;
- *SLEWING*(?*T*1, ?*T*2) : the telescope changes its direction from target ?*T*1 to target ?*T*2.

$<$*VALUE–TOKEN*, **state**(*POINTING–DEVICE*),{*SLEWING*(?*T*, 3C267)}, t_1, $t_2$$>$



DS1: Remote Agent

Remote Agent Experiment: RAX

Remote Agent Experiment

See rax.arc.nasa.gov

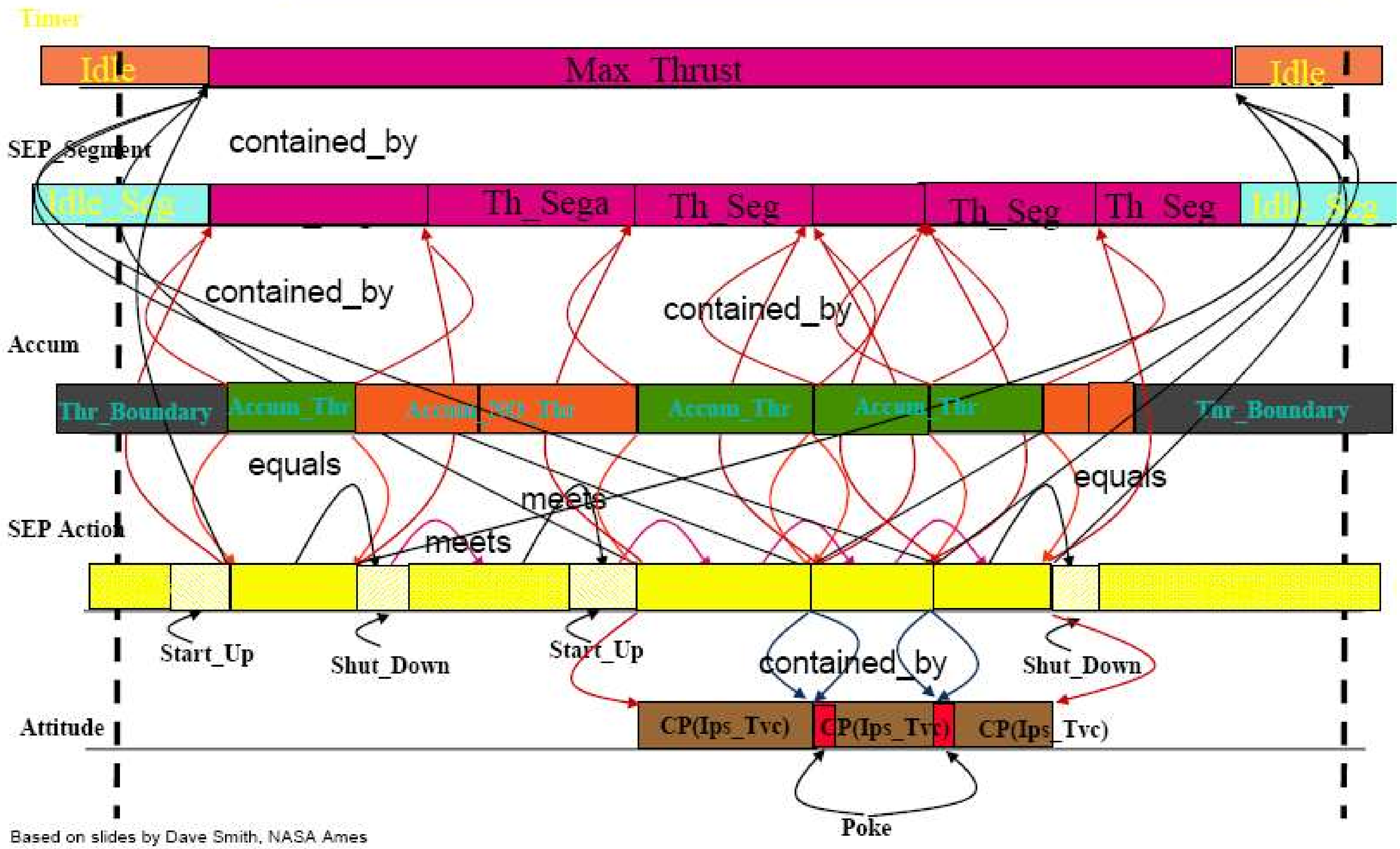
May 17-18th experiment

- 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

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.
  → $10 \leq [T^+(store) - T^-(store)] \leq 30$
- Bread should be eaten within a day of baking.
  → $0 \leq [T^+(baking) - T^-(eating)] \leq 1$ day
- Inequalities, $X^+ < Y^-$, may be expressed as binary interval relations:
  → $-\inf < [X^+ - Y^-] < 0$

Temporal Constraint Networks

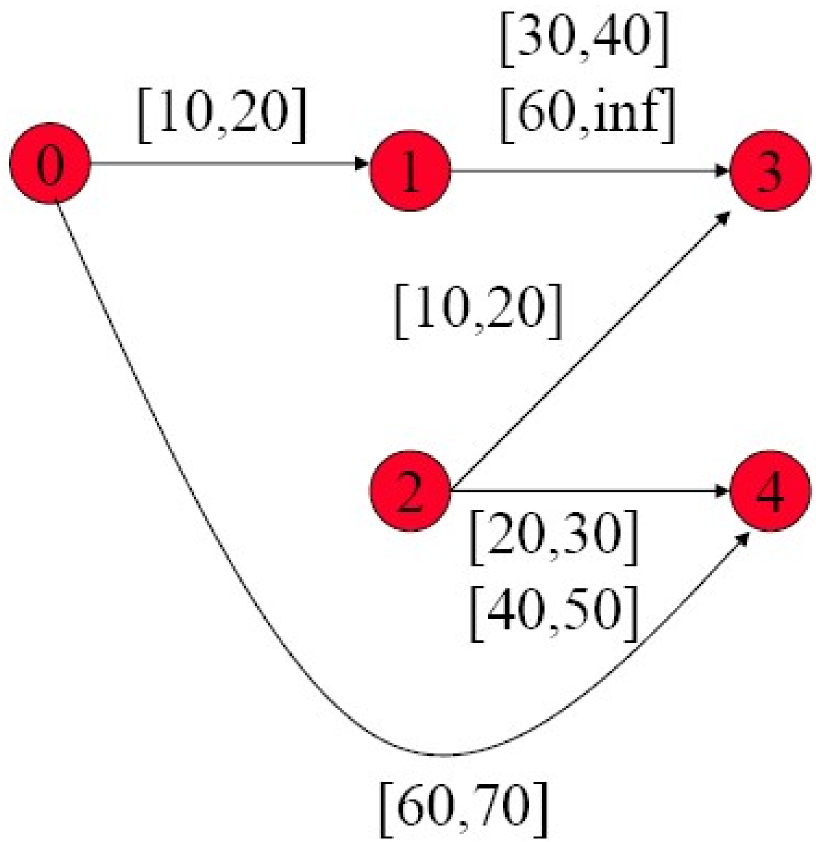
- A set of time points X_i at which events occur.
- Unary constraints

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

- 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

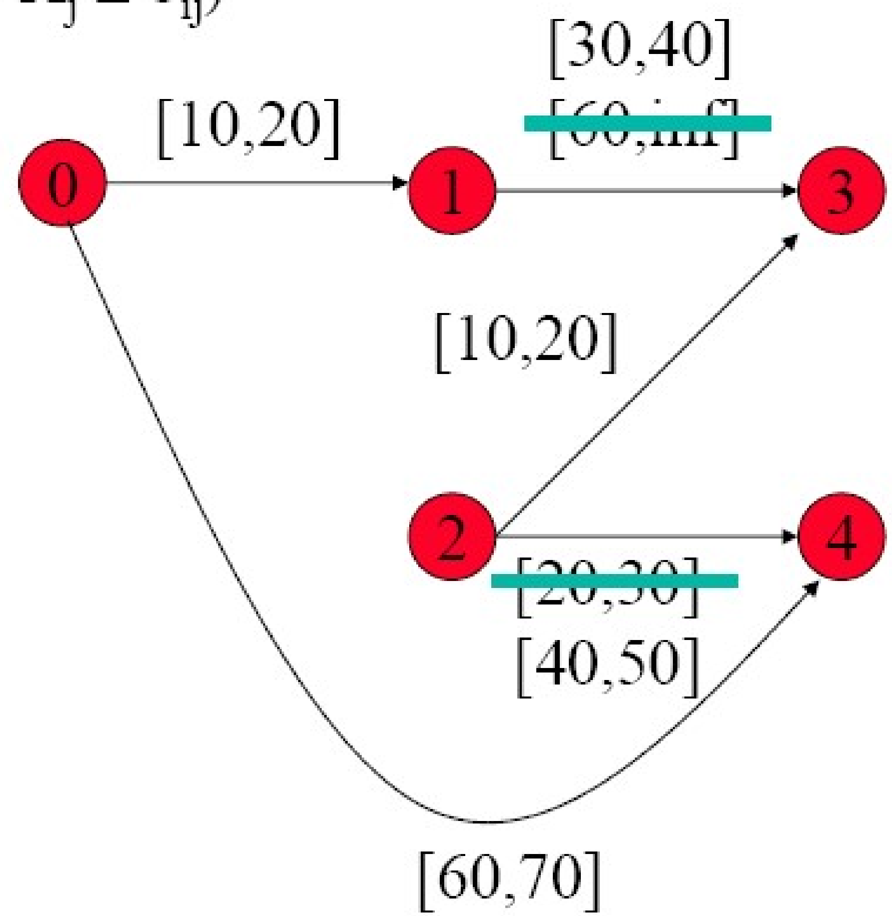
  $(a_0 \leq X_j - X_i \leq b_0)$ ~~or $(a_1 \leq X_j - X_i \leq b_1)$ or . . .~~

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

TCSP Queries

(Dechter, Meiri, Pearl, AIJ91)

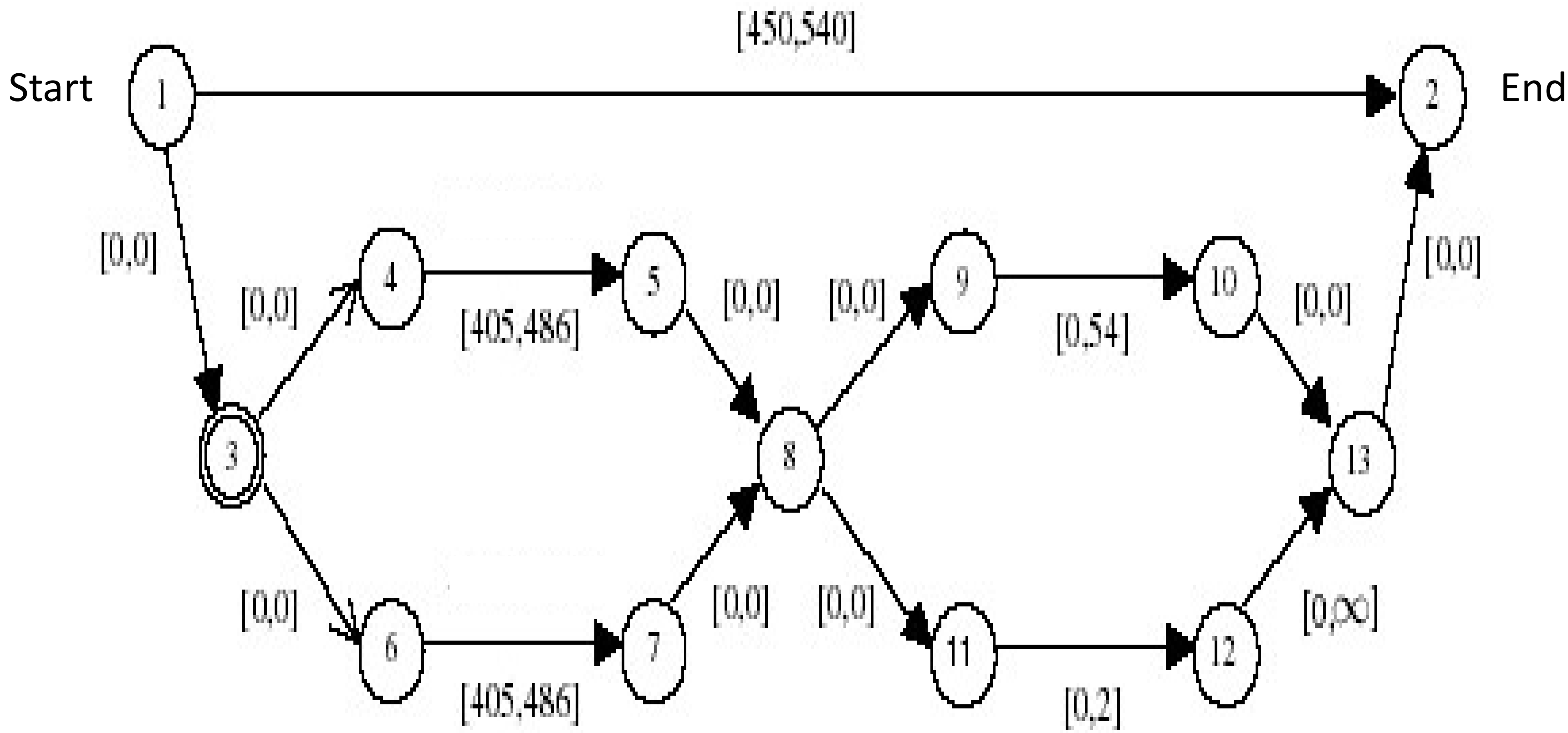
- Is the TCSP consistent?
- What are the feasible times for each X_i?
- What are the feasible durations between each X_i and X_j?
- What is a consistent set of times?
- What are the earliest possible times?
- What are the latest possible times?

TCSP Queries

(Dechter, Meiri, Pearl, AIJ91)

- Is the TCSP consistent? *Planning*
- What are the feasible times for each X_i?
- What are the feasible durations between each X_i and X_j?
- What is a consistent set of times?
- What are the earliest possible times? *Execution*
- What are the latest possible times?

STN example



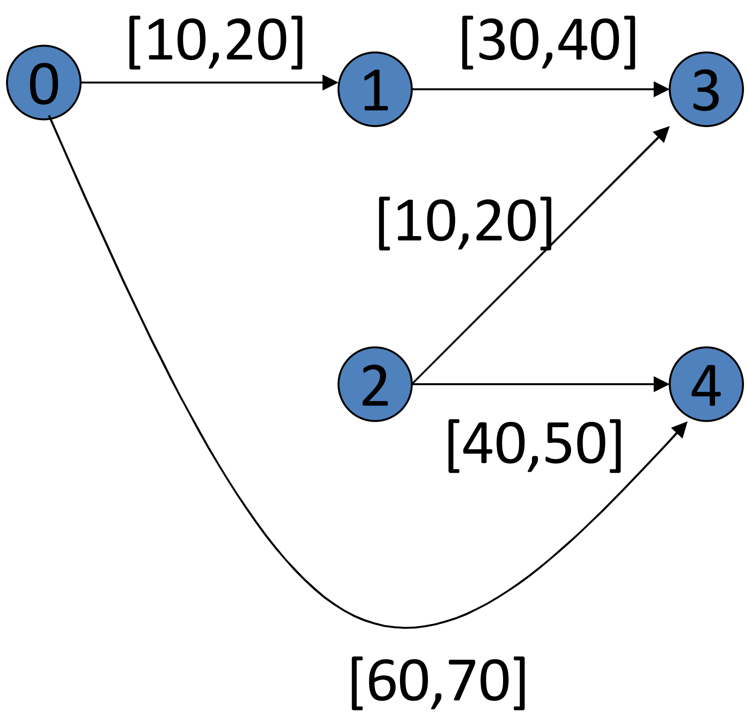
[450,540]
Start
1
2
End
[0,0]
[0,0]
4
5
[405,486]
[0,0]
[0,0]
9
10
[0,54]
[0,0]
[0,0]
3
8
13
[0,0]
6
7
[405,486]
[0,0]
[0,0]
11
12
[0,2]
[0,∞]

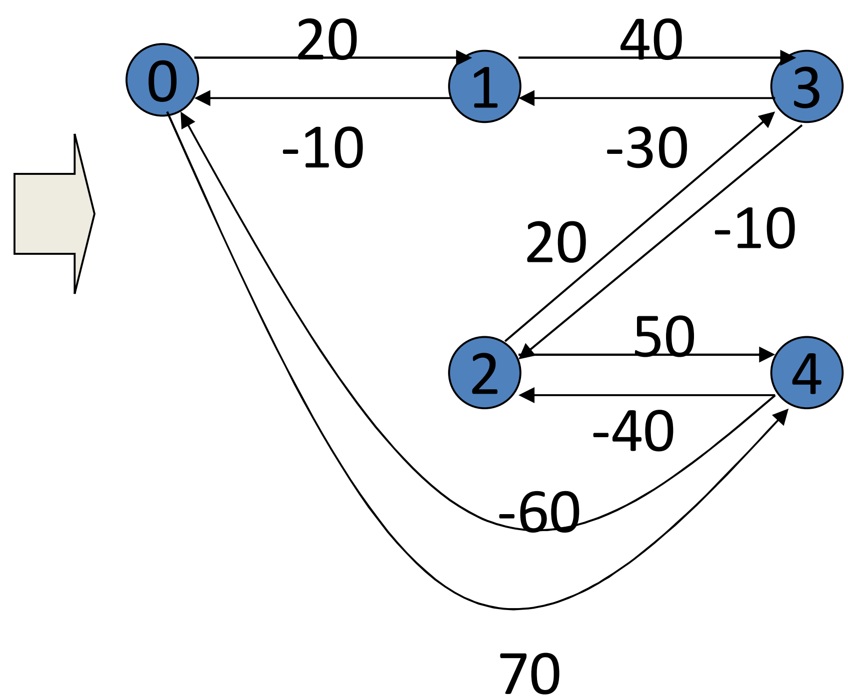
To Query STN Map to Distance Graph $G_d = < V, E_d >$

Edge encodes an upper bound on distance to target from source.

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

$$X_j - X_i \leq b_{ij}$$
$$X_i - X_j \leq - a_{ij}$$





Induced Constraints for G_d

constraint: $i_0 = i, i_1 = \ldots, i_k = j$

$$X_j - X_i \le \sum_{j=1}^{k} a_{i_{j-1}, i_j}$$

→Intersected path constraints:

$$X_j - X_i \le d_{ij}$$

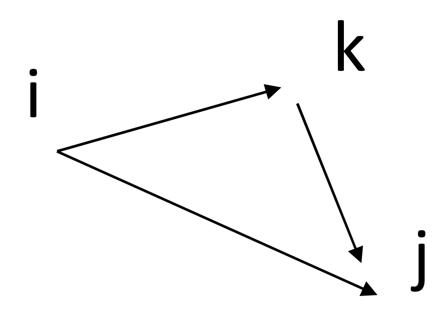
where d_{ij} is the shortest path from i to j

Compute Intersected Paths by All Pairs Shortest Path

(e.g., Floyd-Warshall's algorithm)

1. for i := 1 to n do $d_{ii} \leftarrow 0$;
2. for i, j := 1 to n do $d_{ij} \leftarrow a_{ij}$;

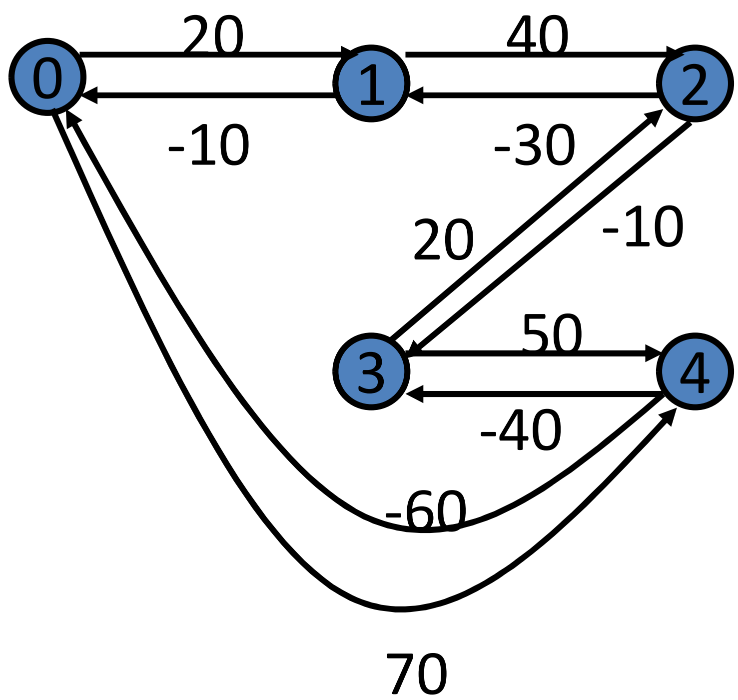
3. for k := 1 to n do
4.   for i, j := 1 to n do
5. $d_{ij} \leftarrow \min\{d_{ij}, d_{ik} + d_{kj}\}$;



Shortest Paths of G_d

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph



STN Minimum Network

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph

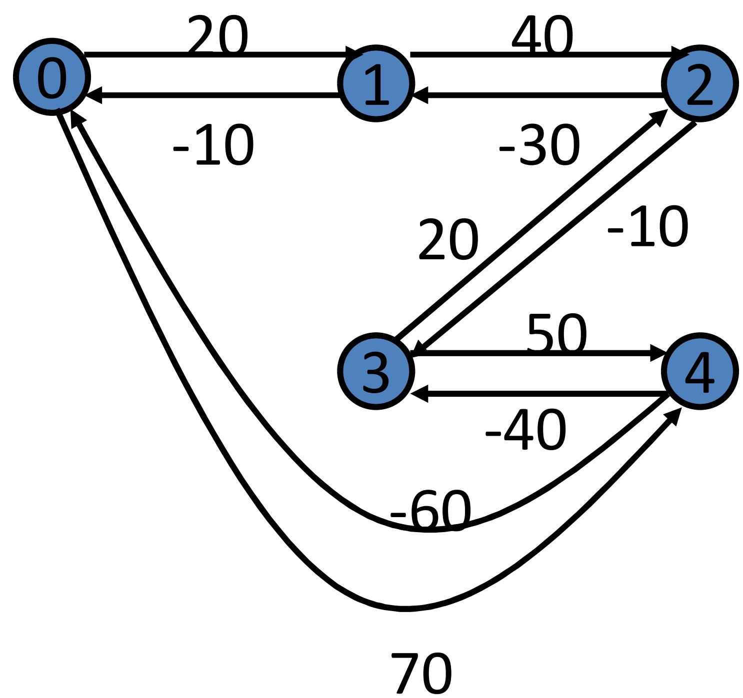
	0	1	2	3	4
0	[0]	[10,20]	[40,50]	[20,30]	[60,70]
1	[-20,-10]	[0]	[30,40]	[10,20]	[50,60]
2	[-50,-40]	[-40,-30]	[0]	[-20,-10]	[20,30]
3	[-30,-20]	[-20,-10]	[10,20]	[0]	[40,50]
4	[-70,-60]	[-60,-50]	[-30,-20]	[-50,-40]	[0]

STN minimum network

Test Consistency:
No Negative Cycles

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph

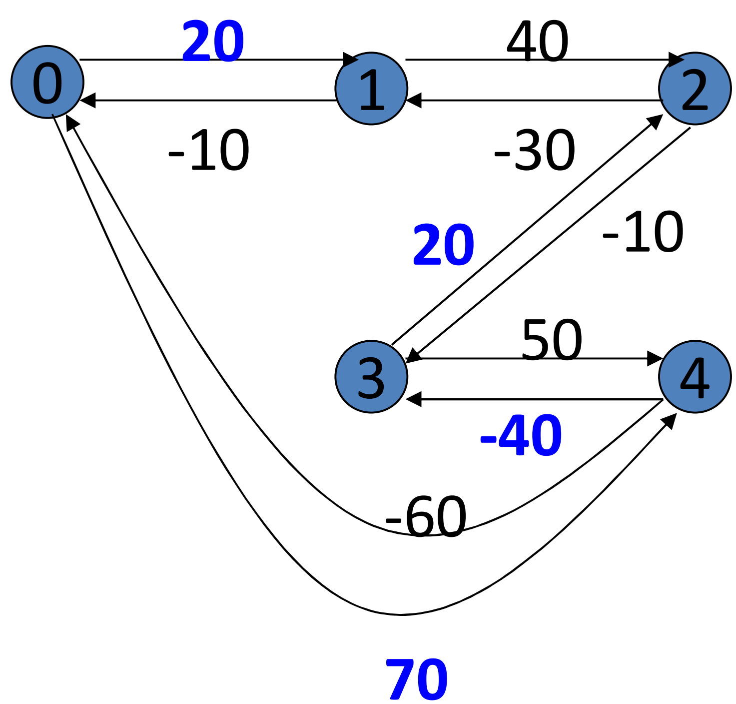


Latest Solution

Node 0 is the reference.

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph

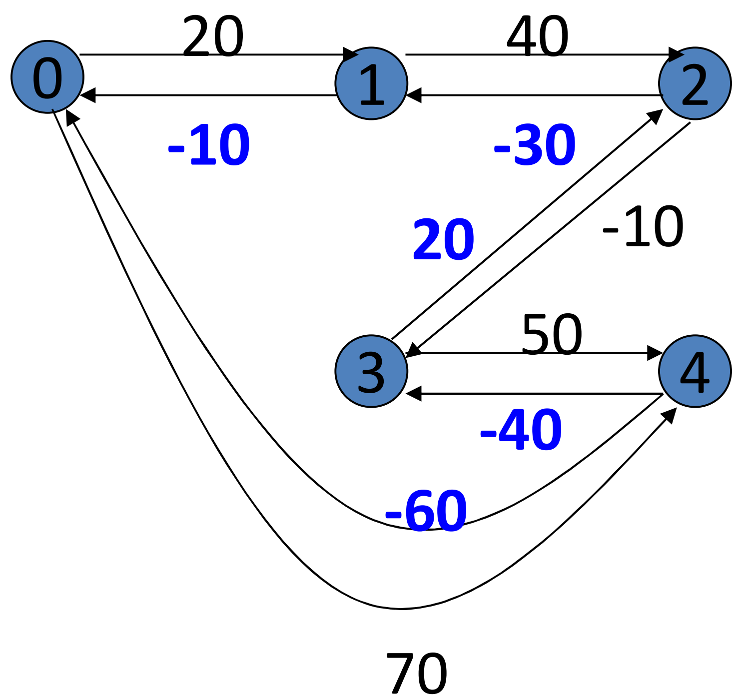


Earliest Solution

Node 0 is the reference.

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph



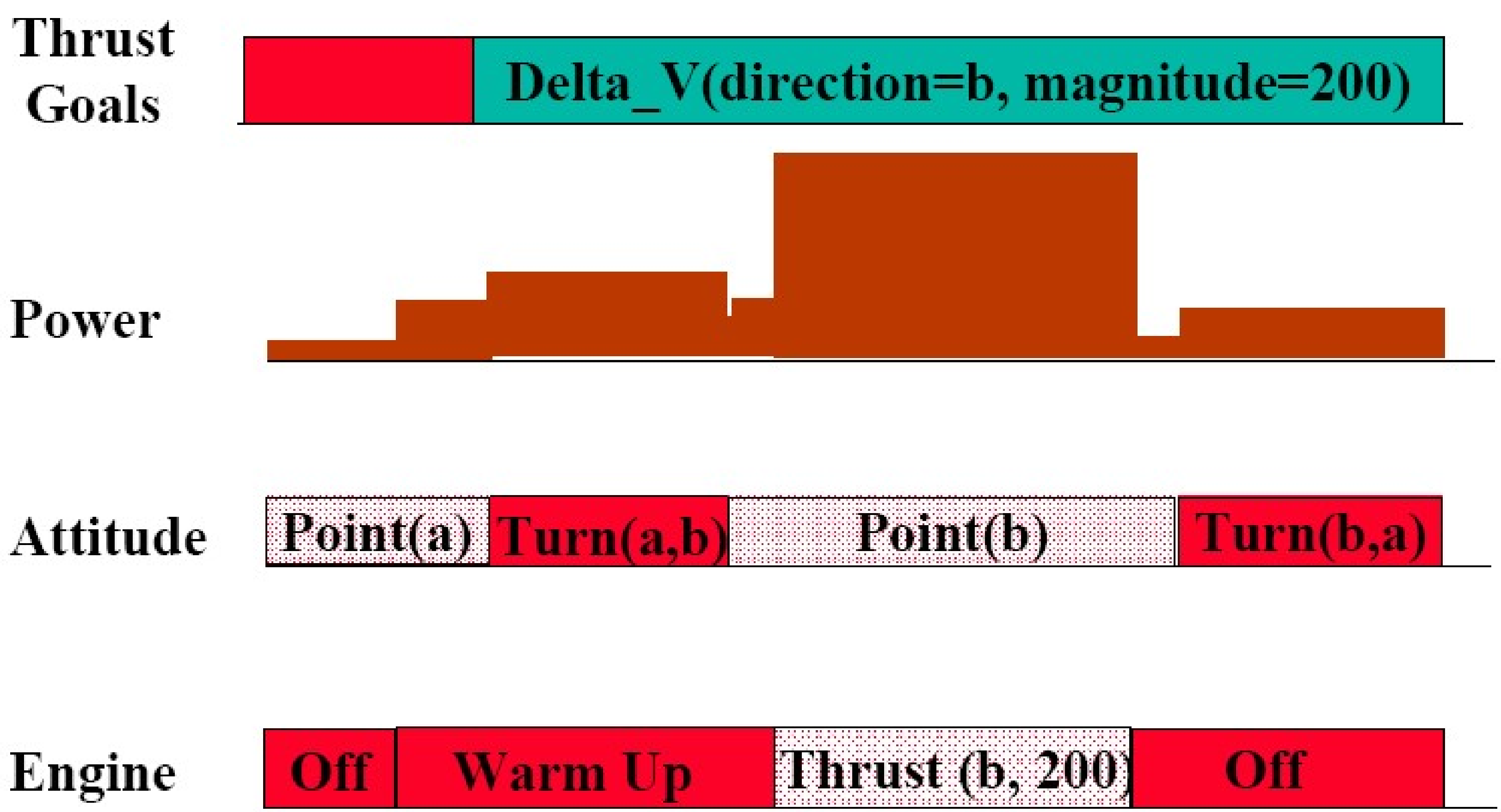
Feasible Values

	0	1	2	3	4
0	0	20	50	30	70
1	-10	0	40	20	60
2	-40	-30	0	-10	30
3	-20	-10	20	0	50
4	-60	-50	-20	-40	0

d-graph

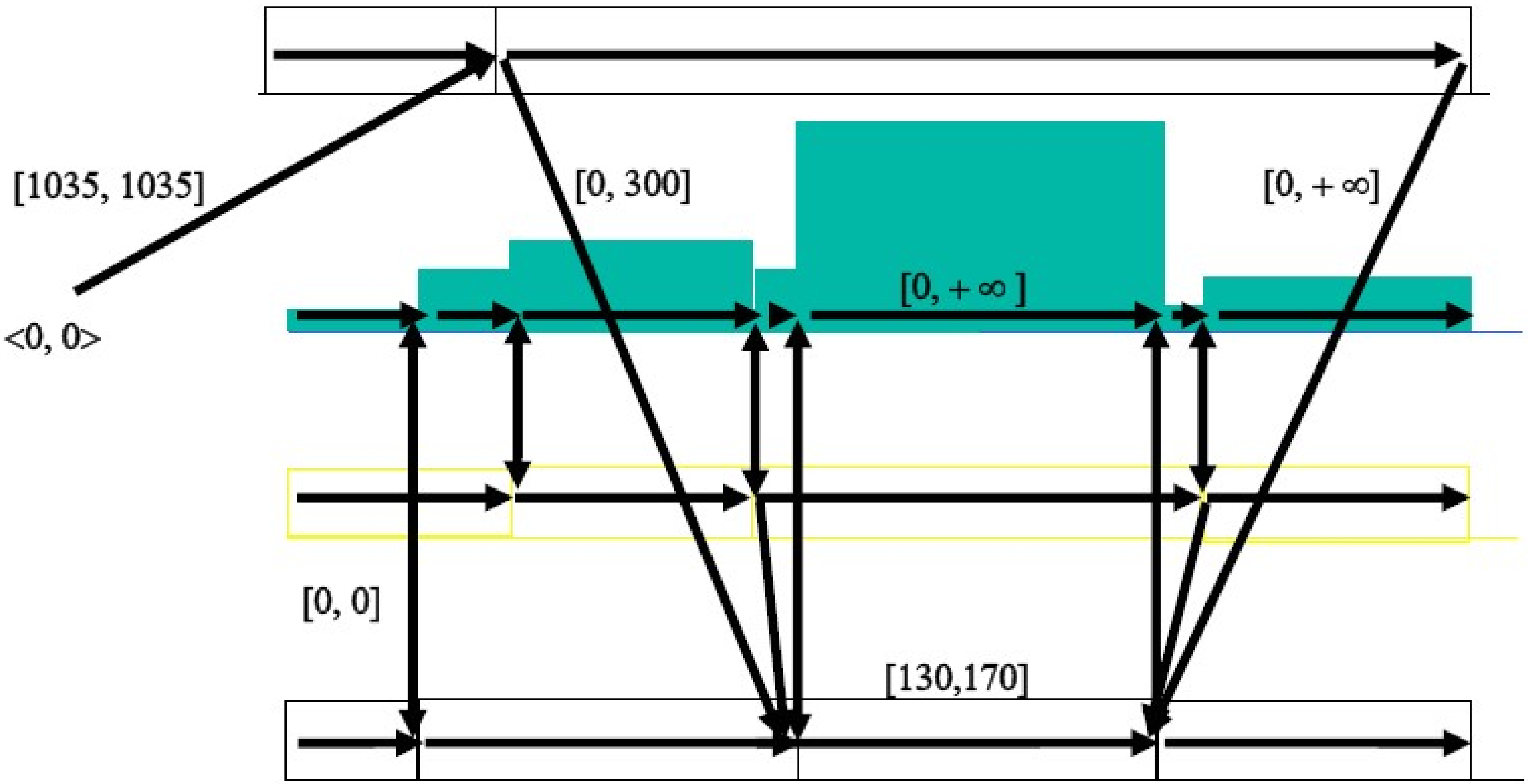
- X_1 in [10, 20]
- X_2 in [40, 50]
- X_3 in [20, 30]
- X_4 in [60, 70]

A Complete CBI-Plan is a STN



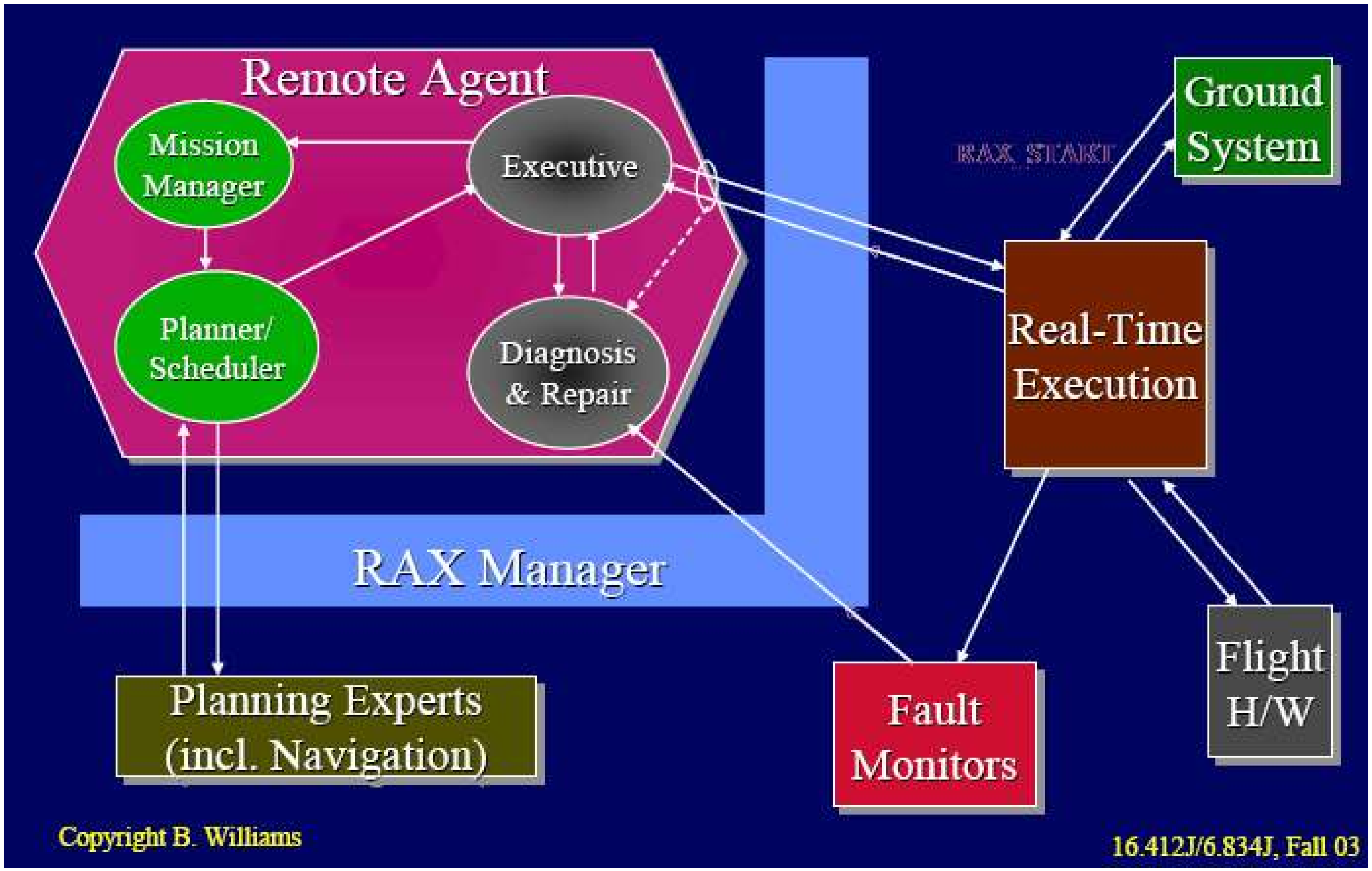
Based on slides by Dave Smith, NASA Ames

A Complete CBI-Plan is a STN

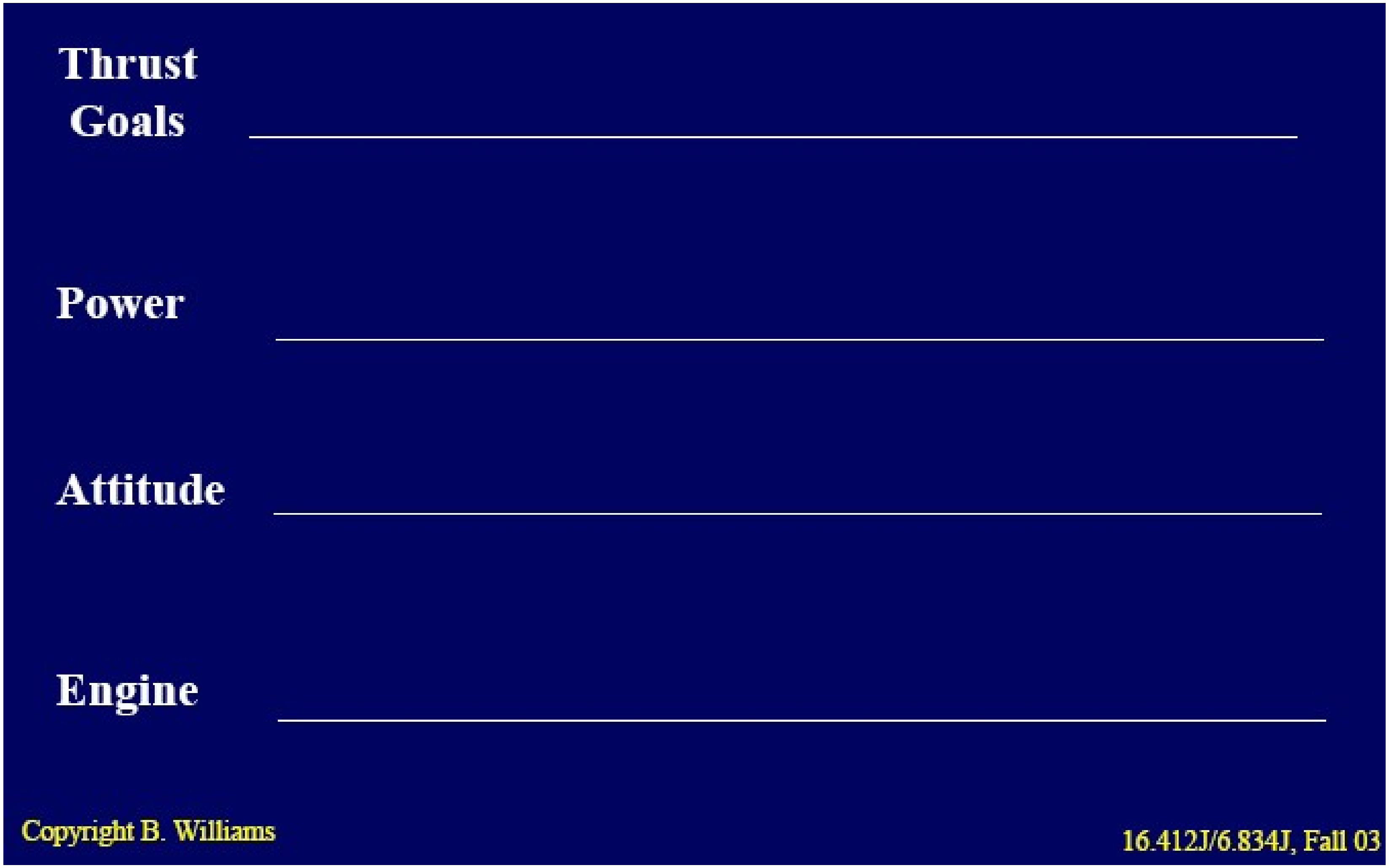


Based on slides by Dave Smith, NASA Ames

Remote Agent

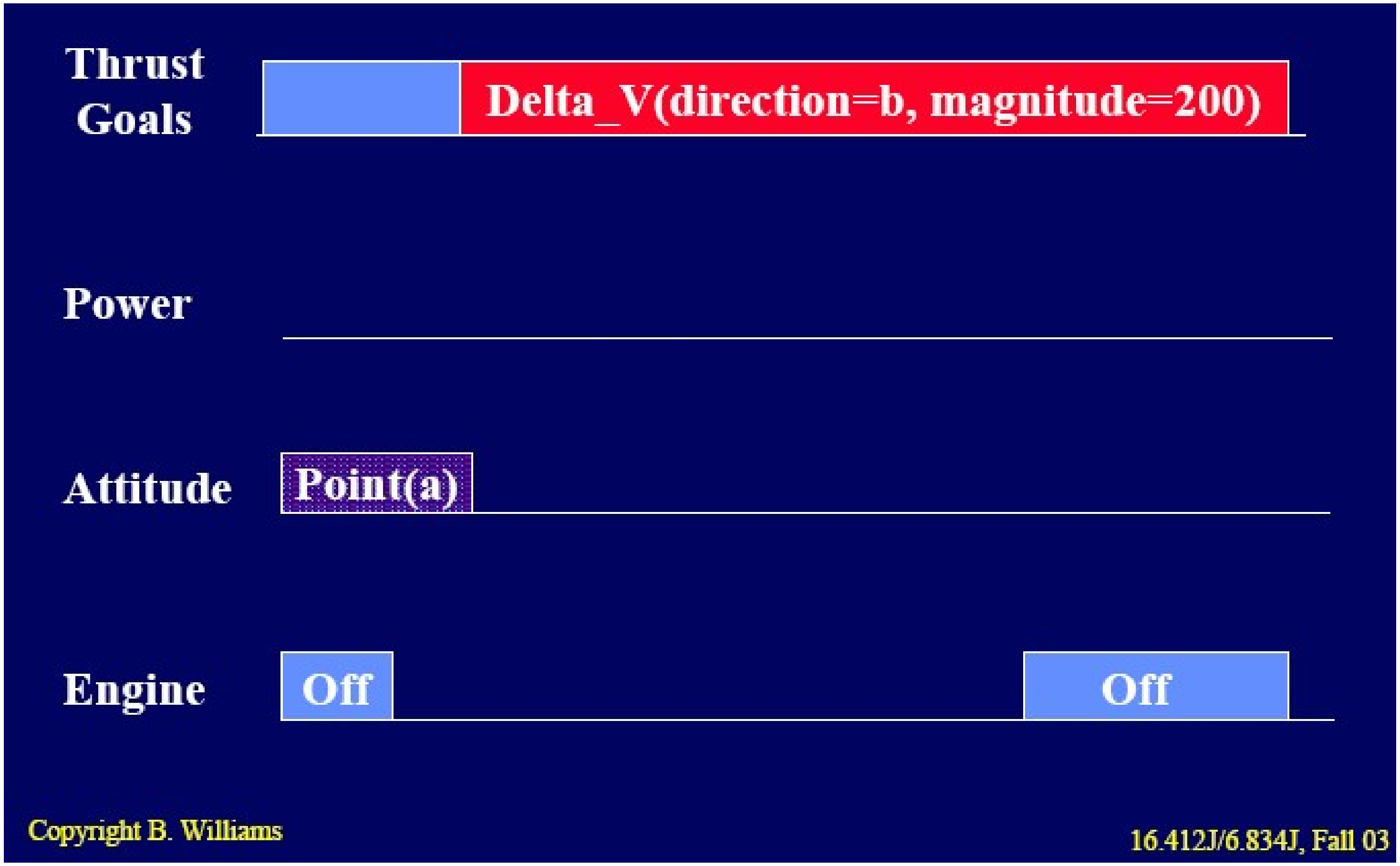


Remote Agent



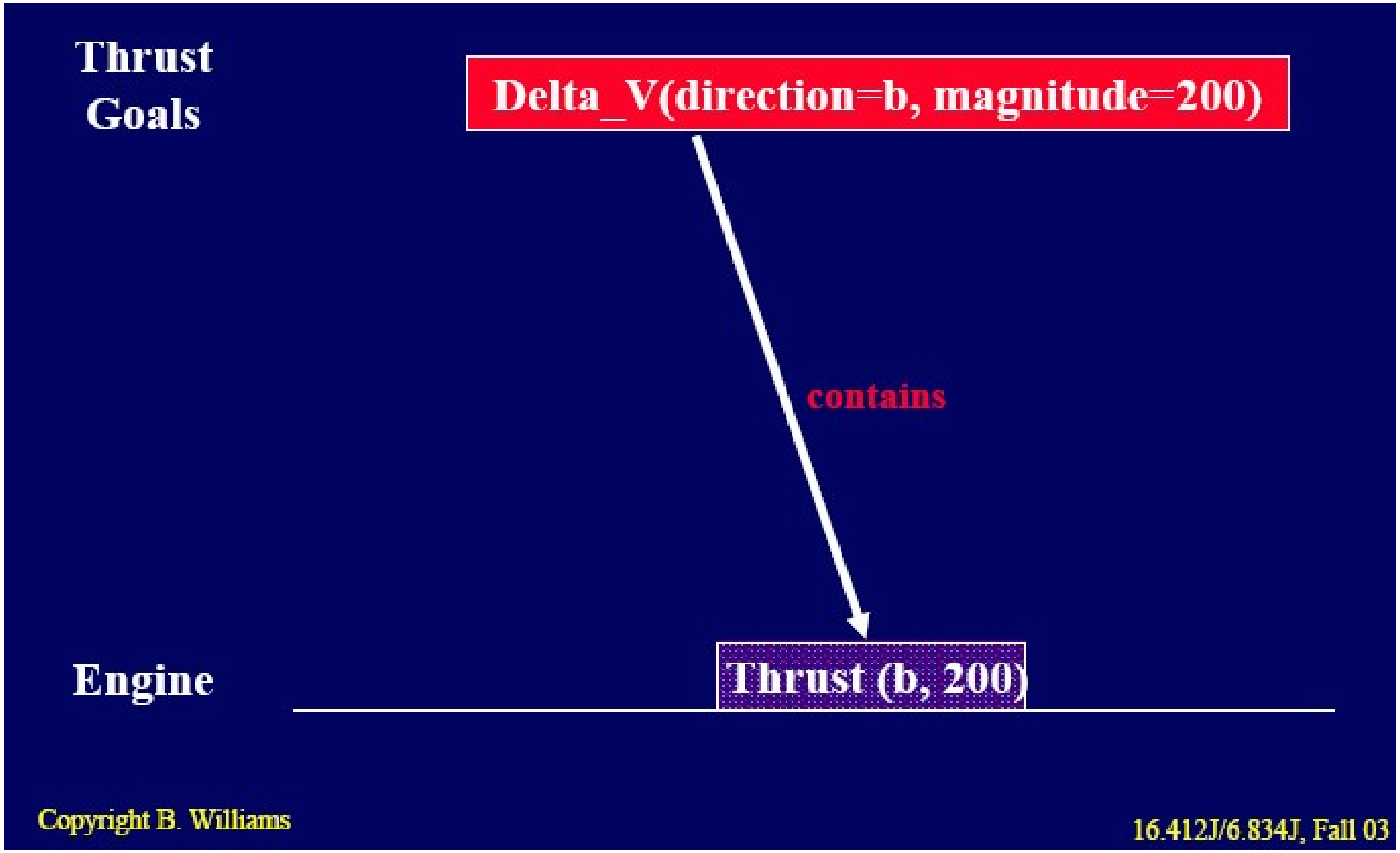
Remote Agent

- Mission Manager



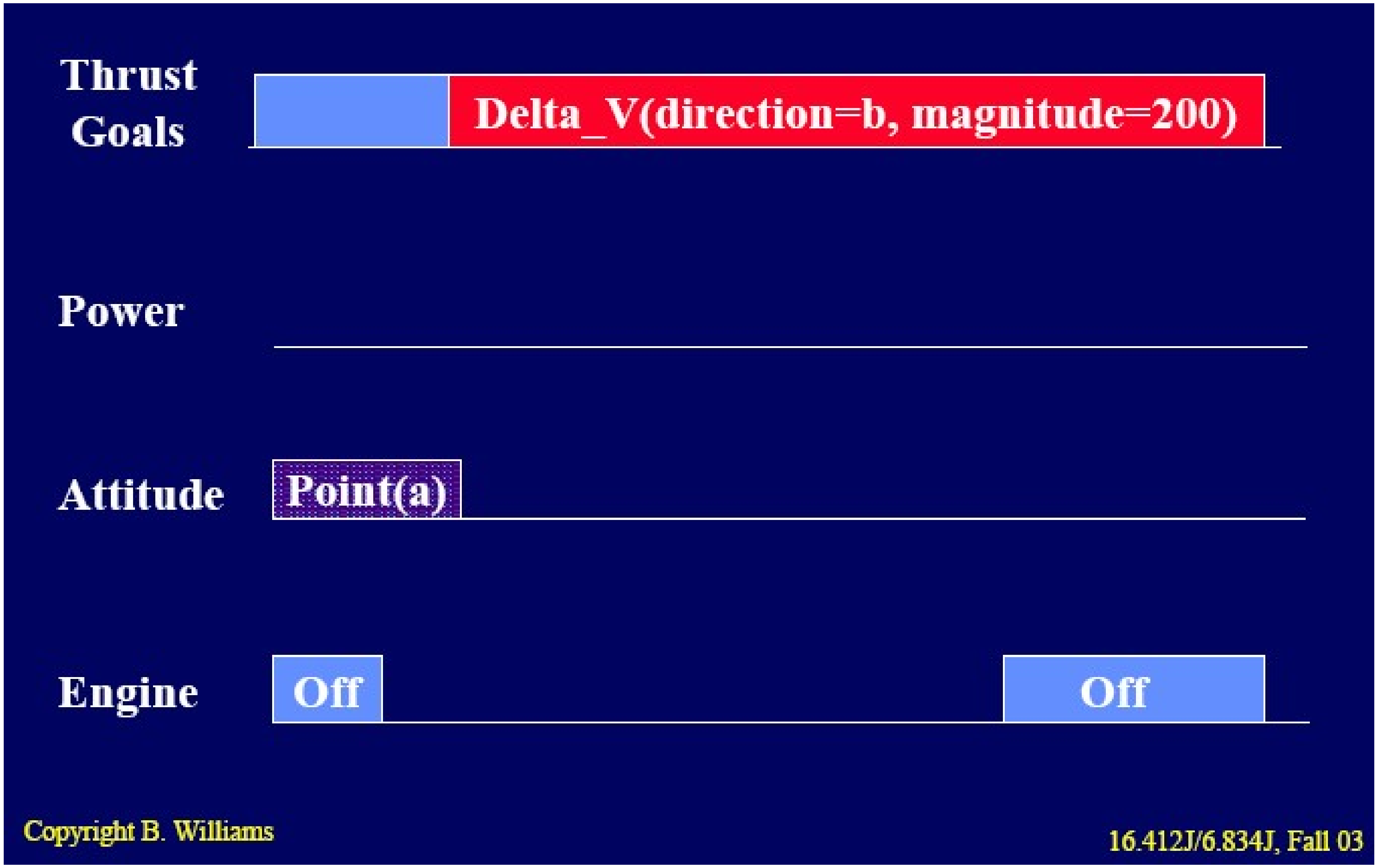
Remote Agent

- Constraints:



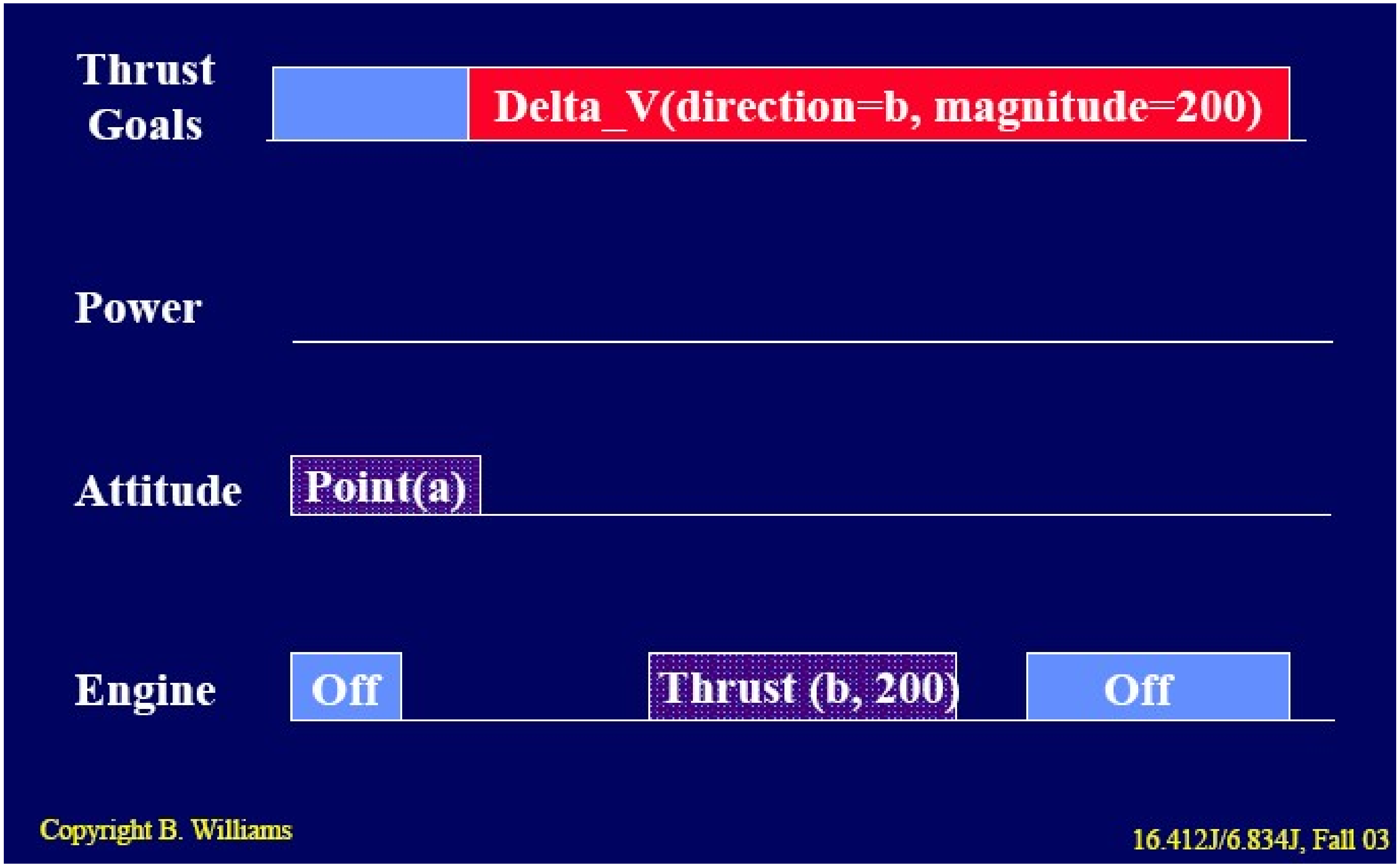
Remote Agent

- Planner starts



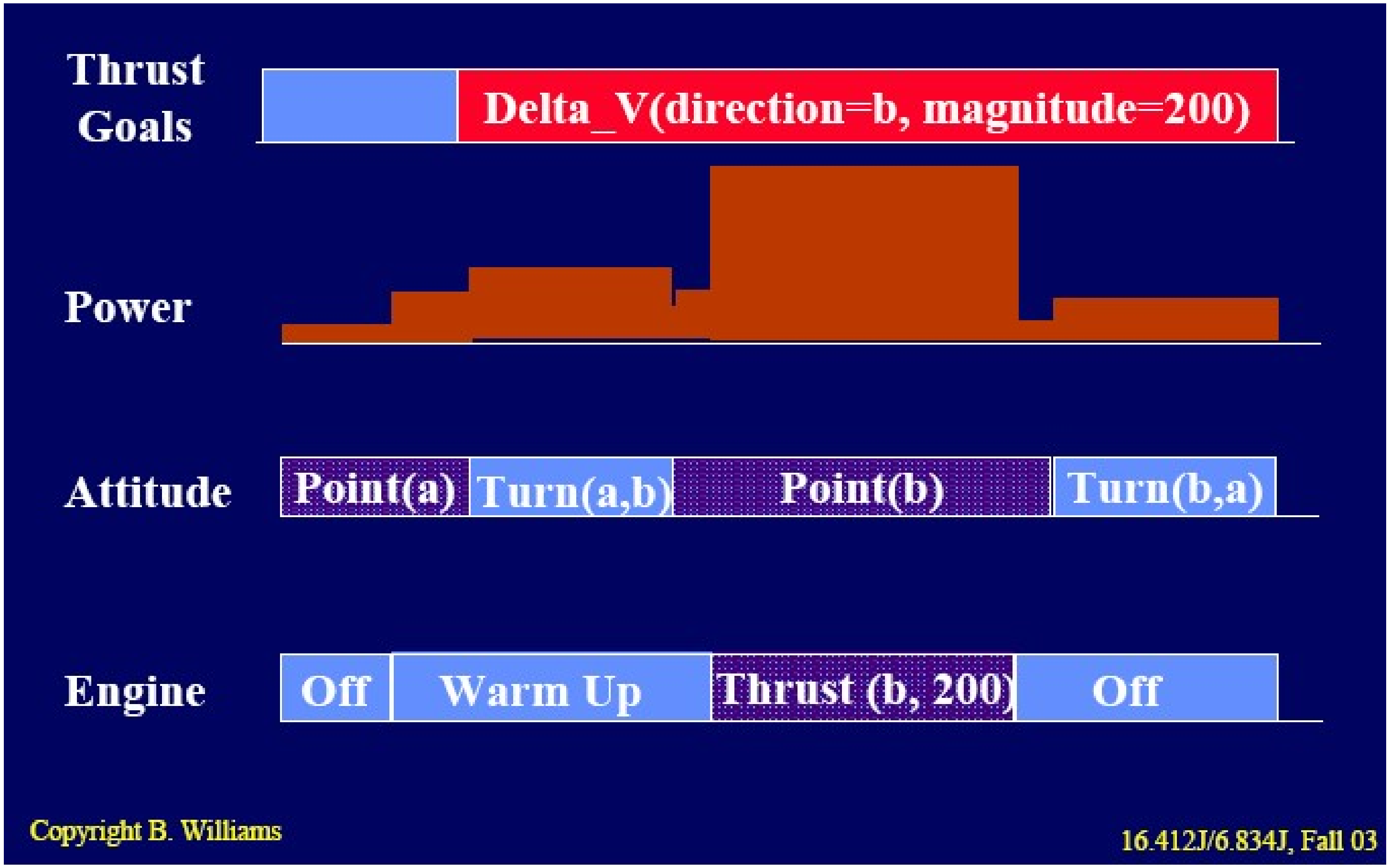
Remote Agent

- Planning



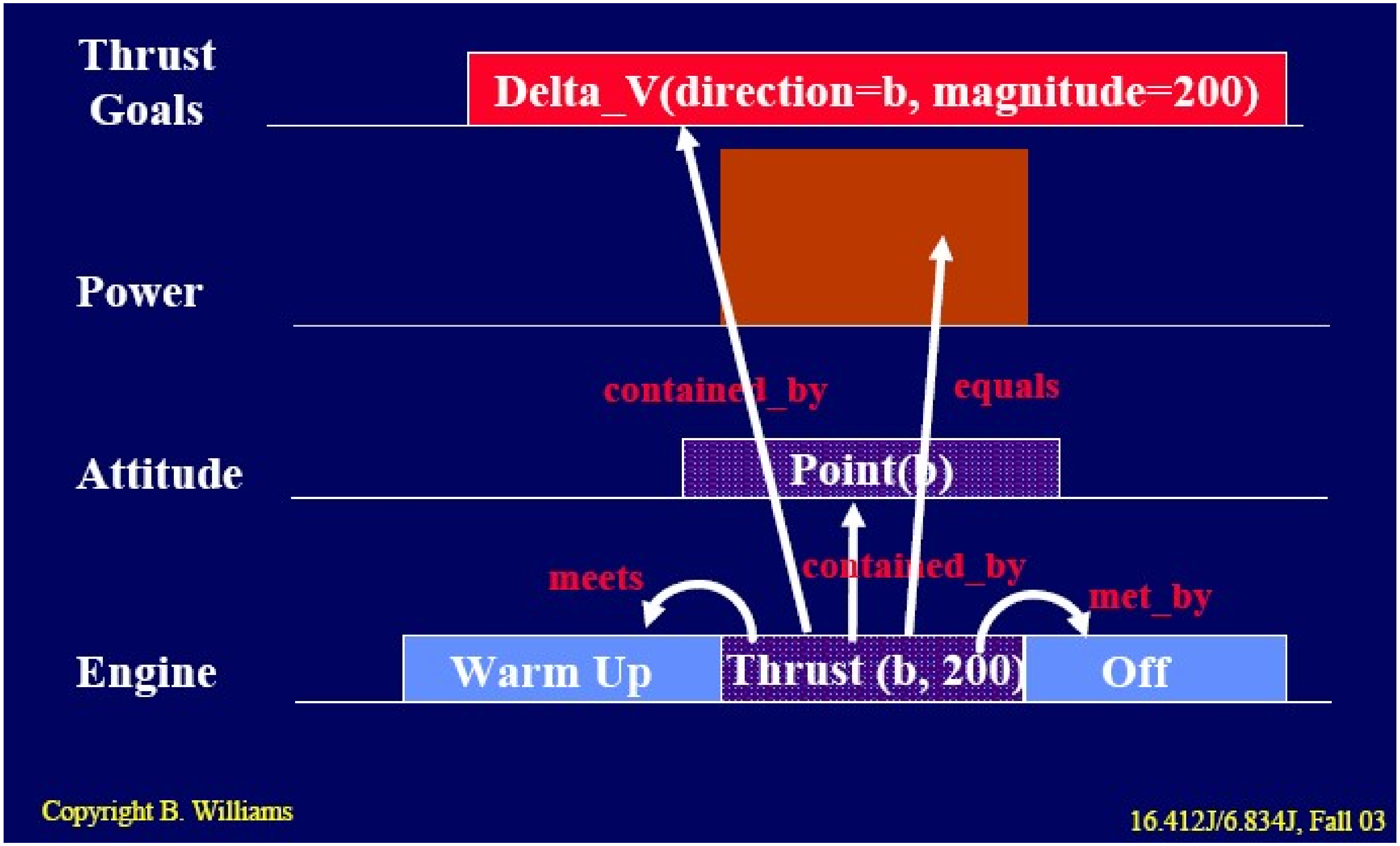
Remote Agent

- Final Plan



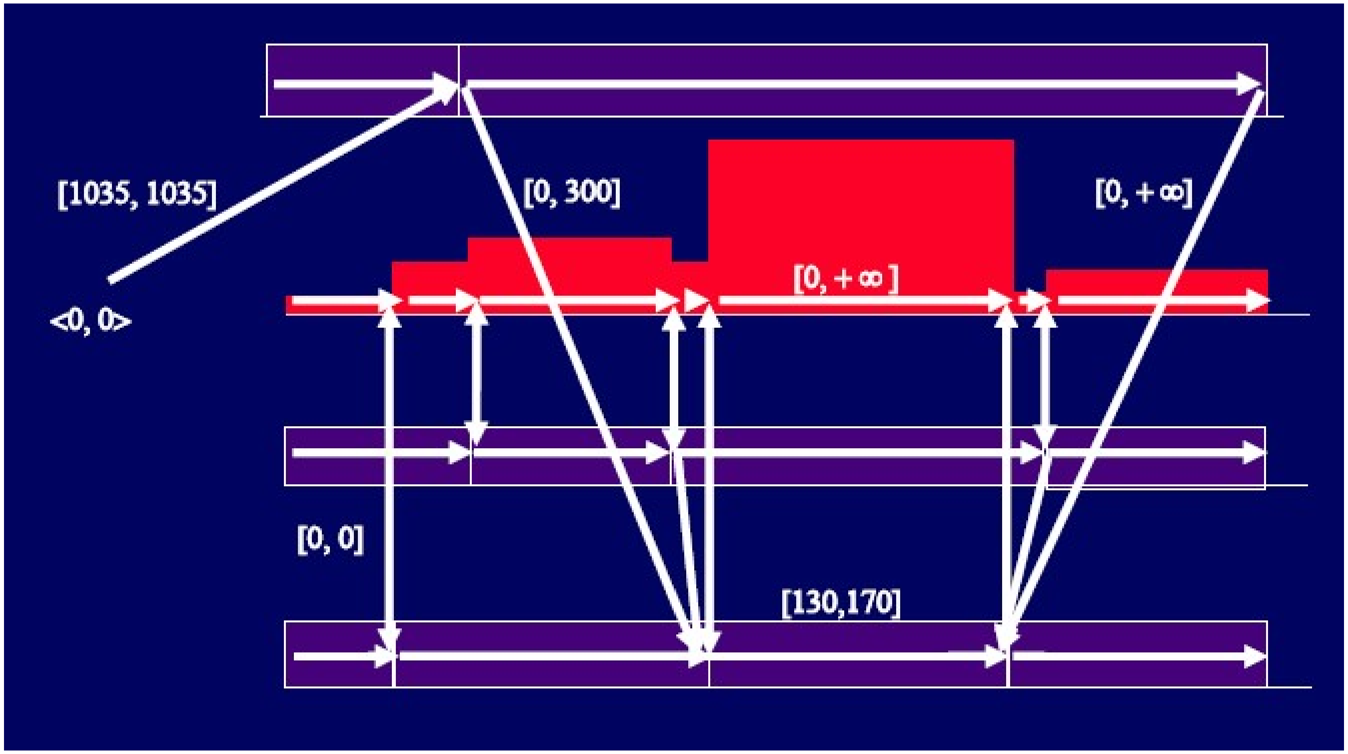
Remote Agent

- Constraints



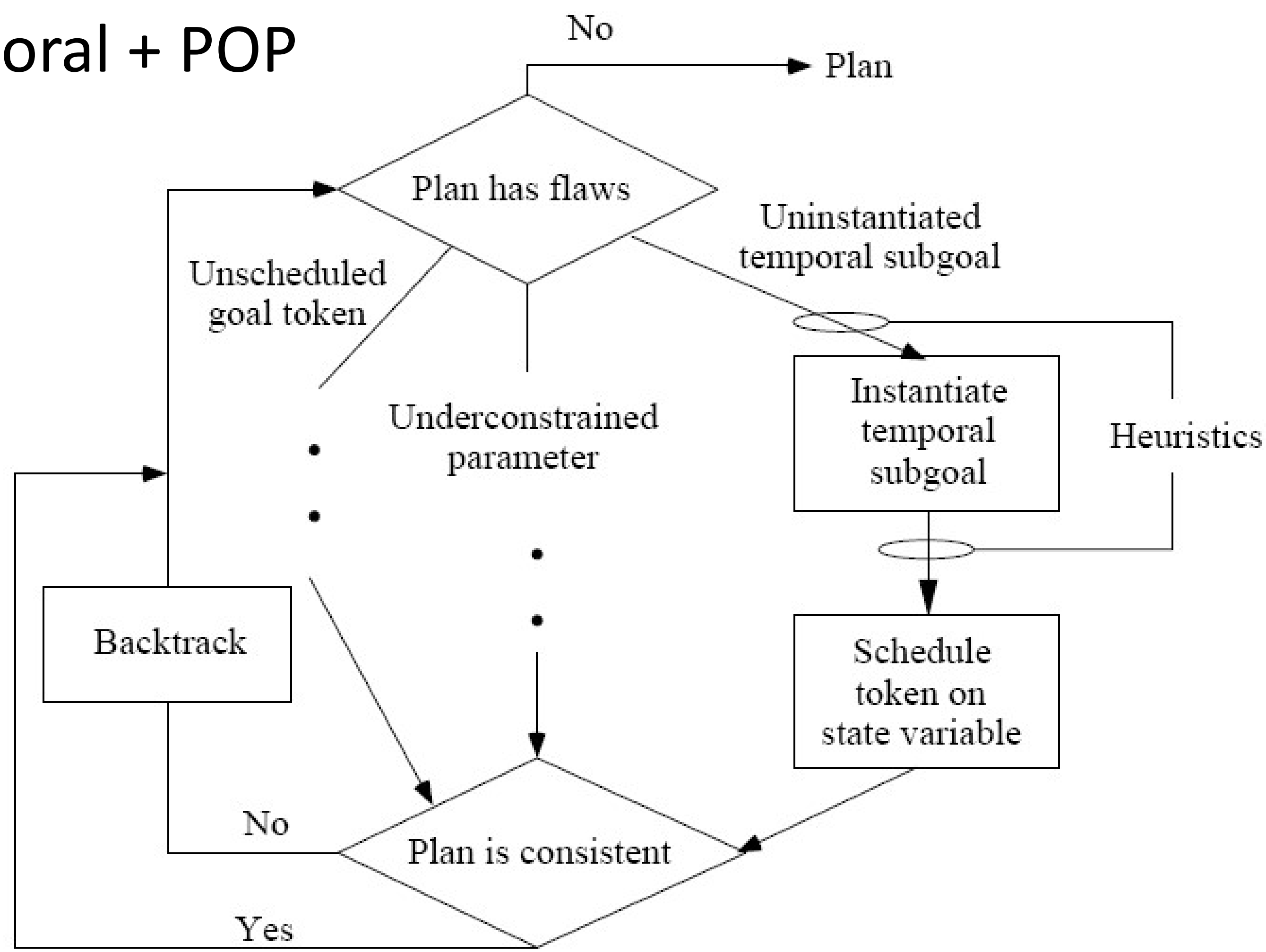
Remote Agent

- Flexible Temporal Plan through least commitment



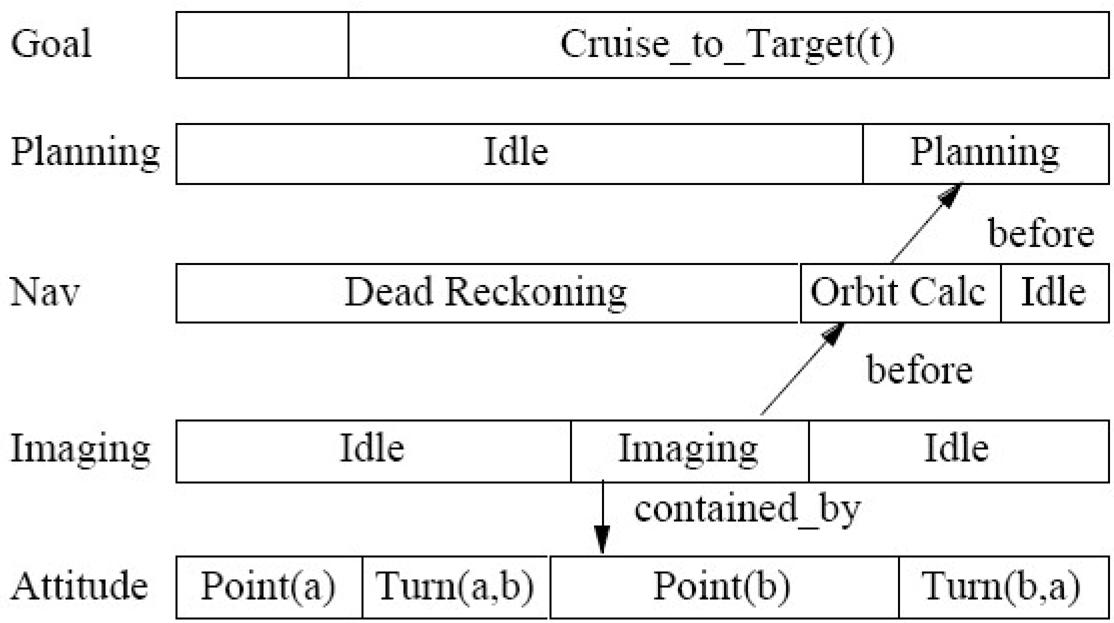
Remote Agent

- Planning
  - Temporal + POP



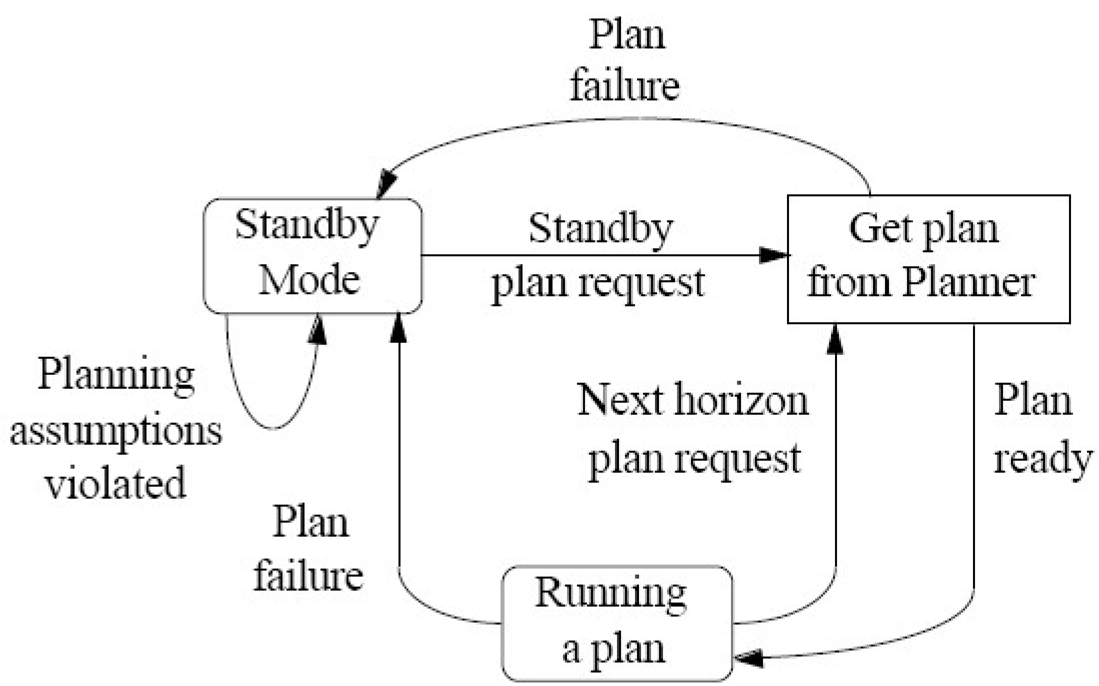
Remote Agent

- Planning to plan



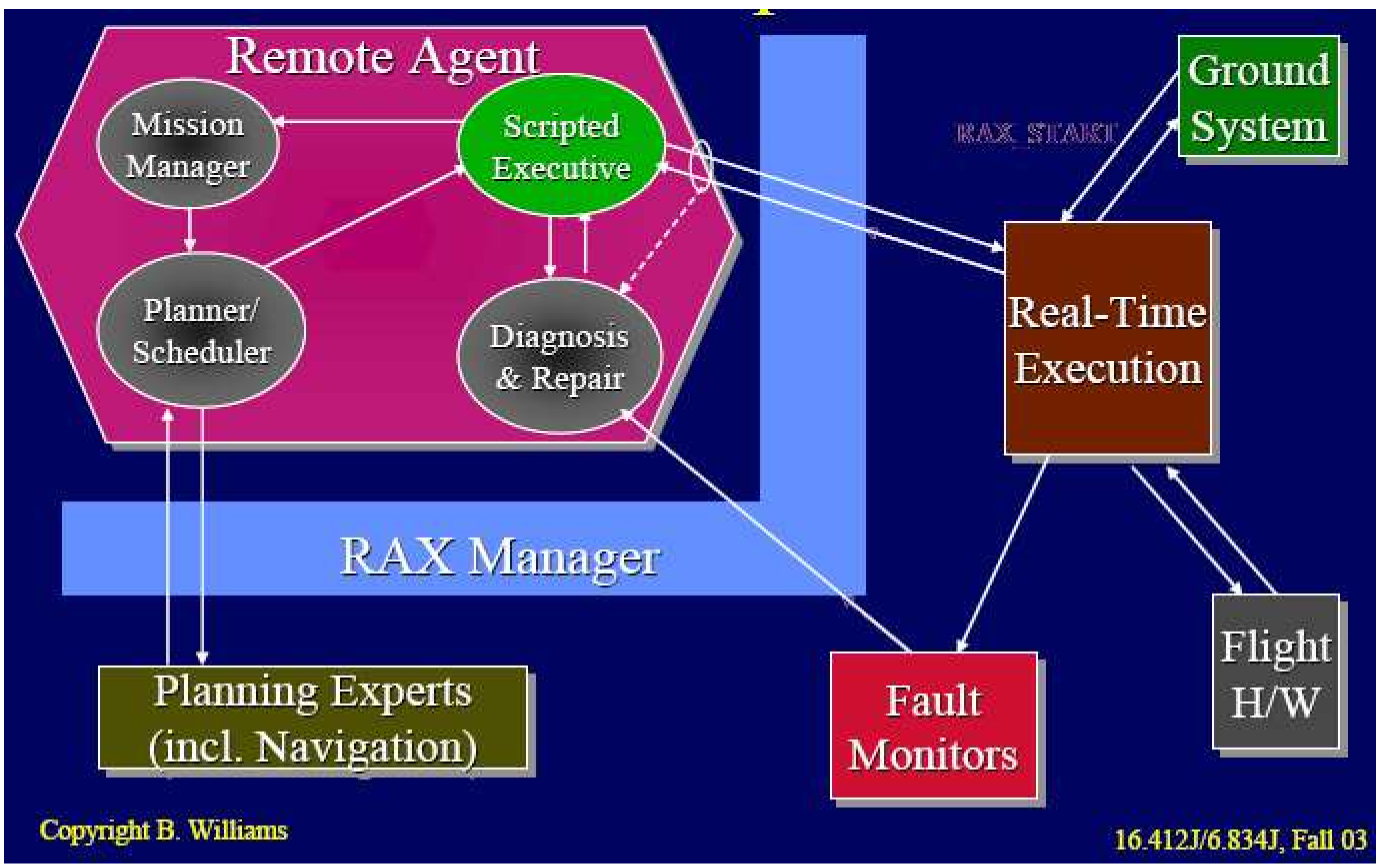
Remote Agent

- Periodic planning and replanning



RAX: Remote Agent

- Executive system dispatch tasks

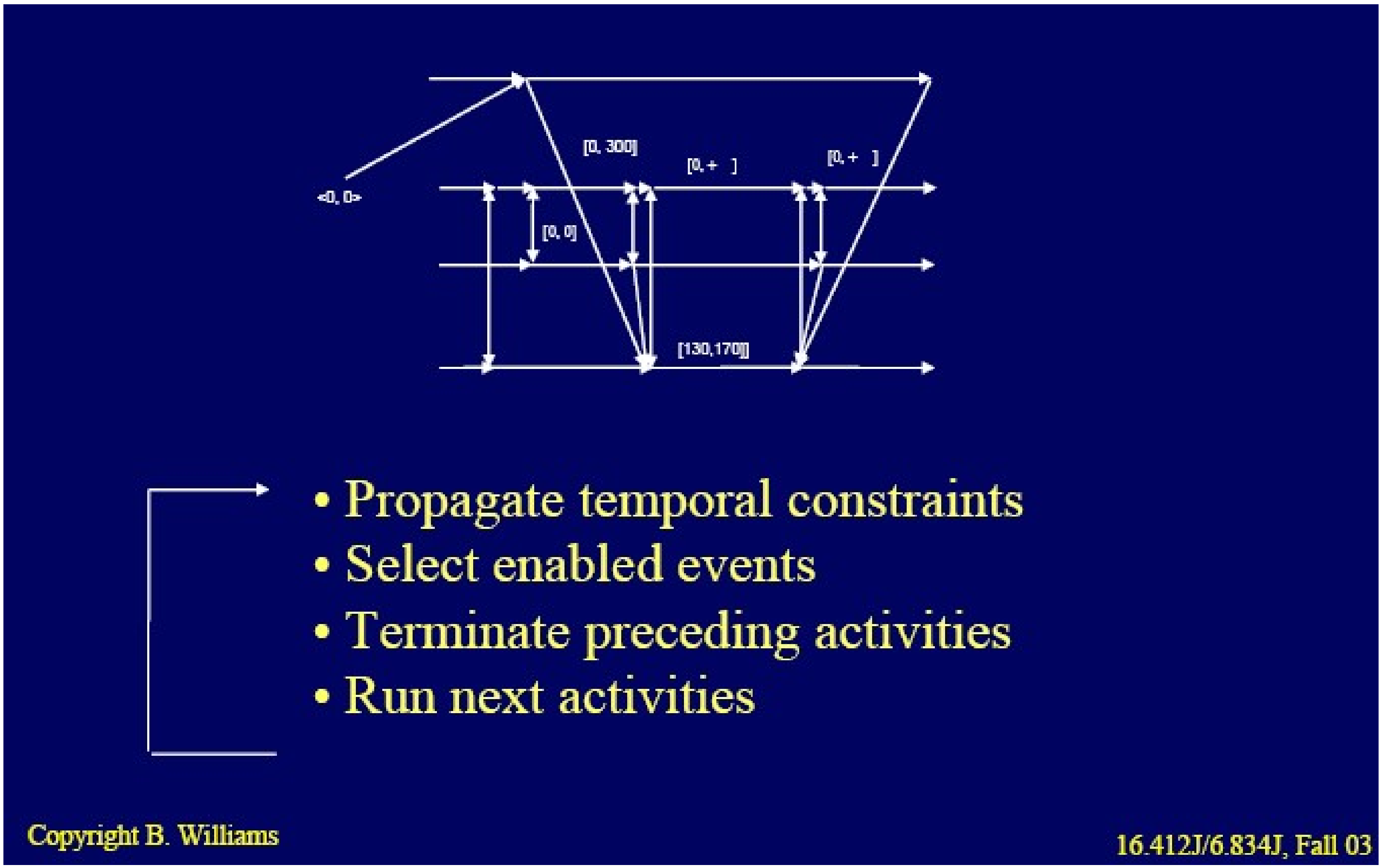


Remote Agent

- 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

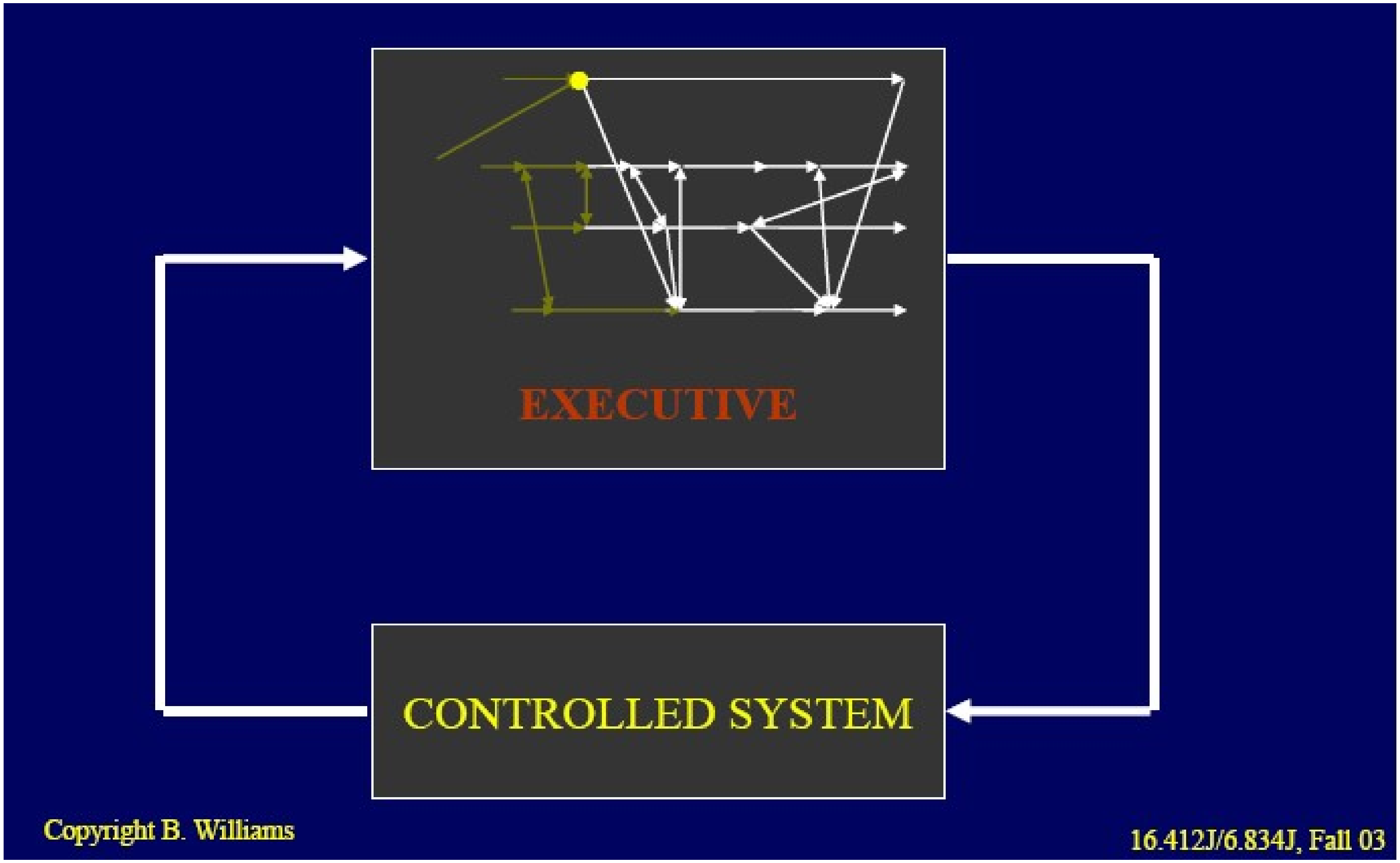
Remote Agent

- Executing Flexible Plans



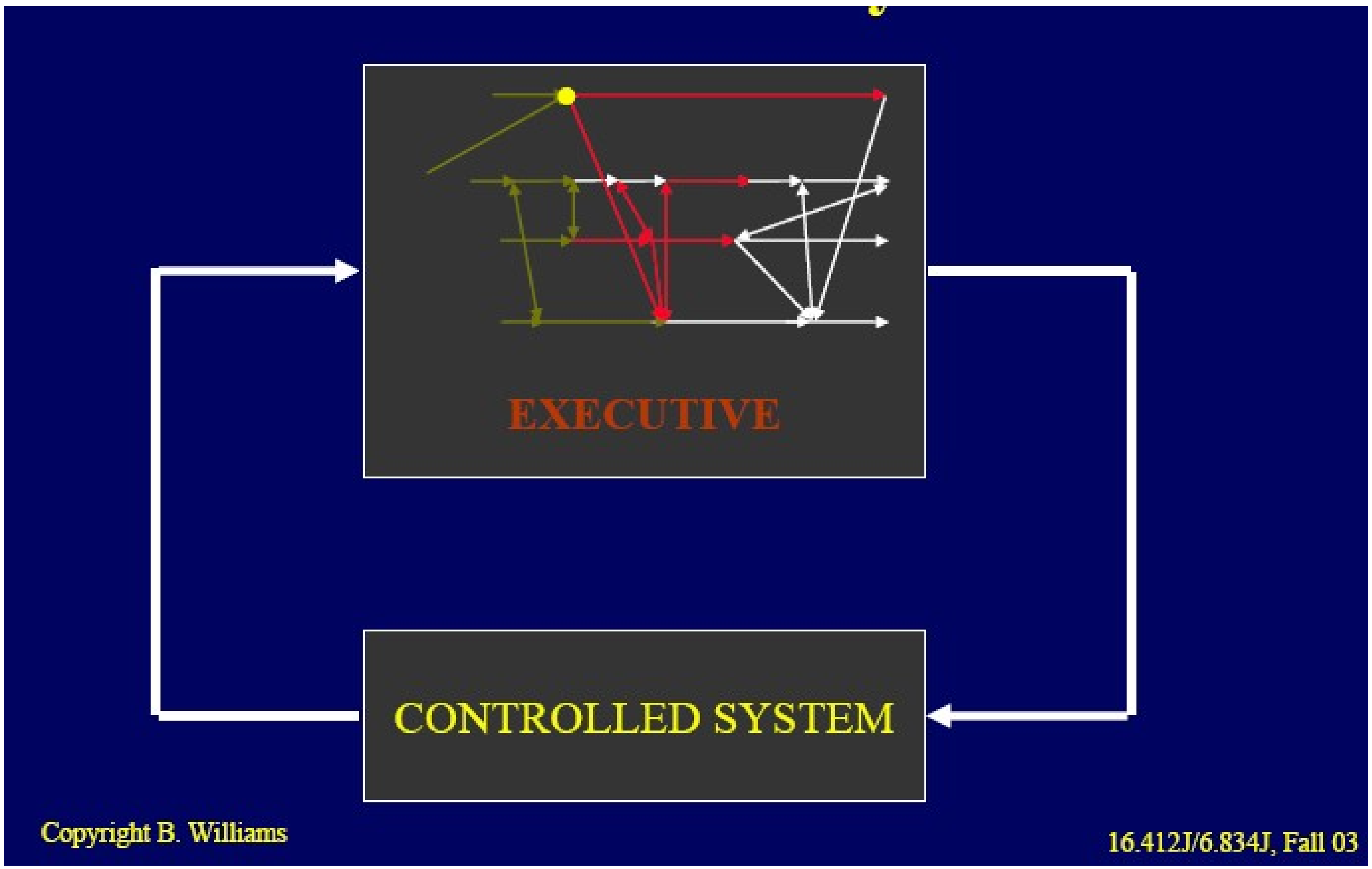
Remote Agent

- Constraint propagation can be costly



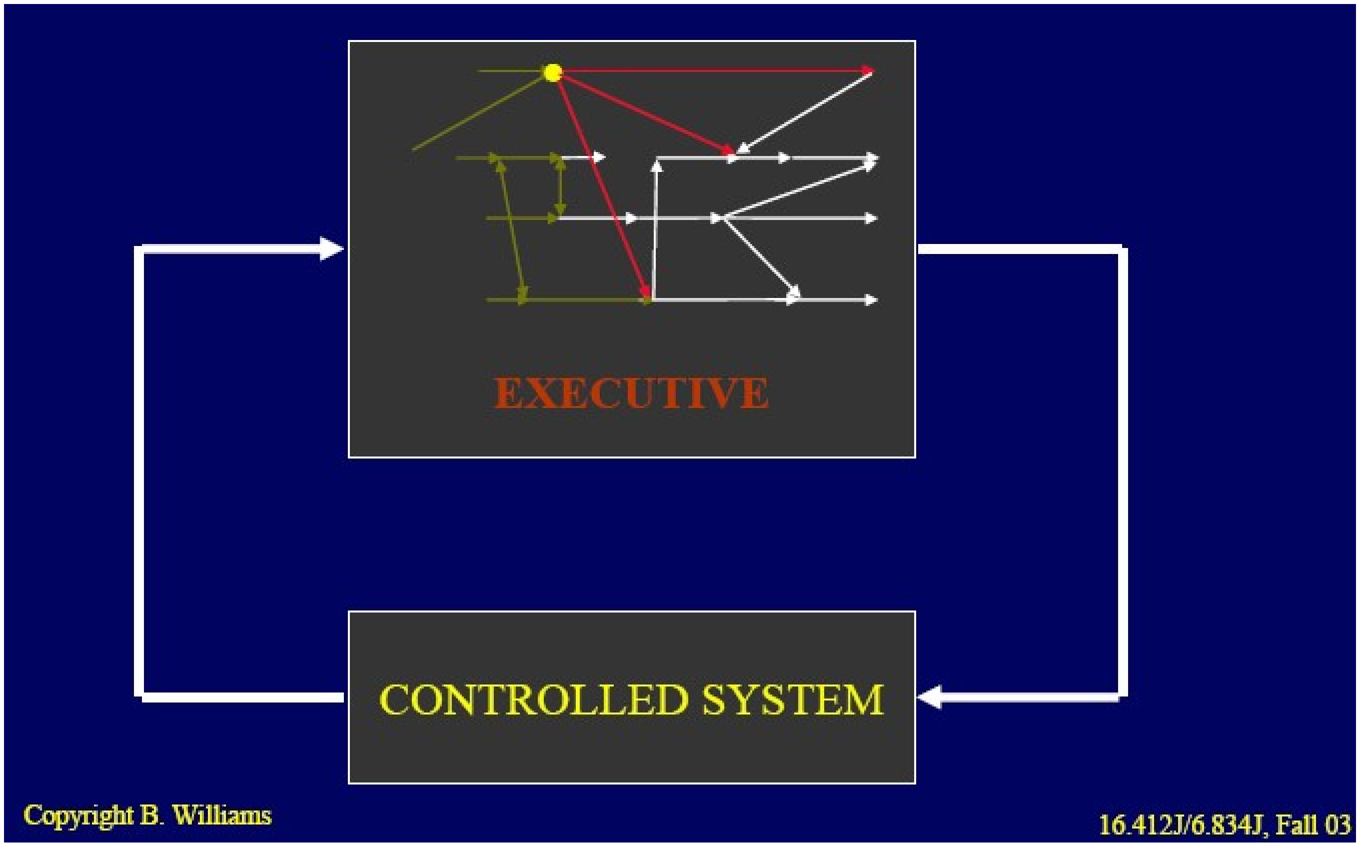
Remote Agent

- Constraint propagation can be costly



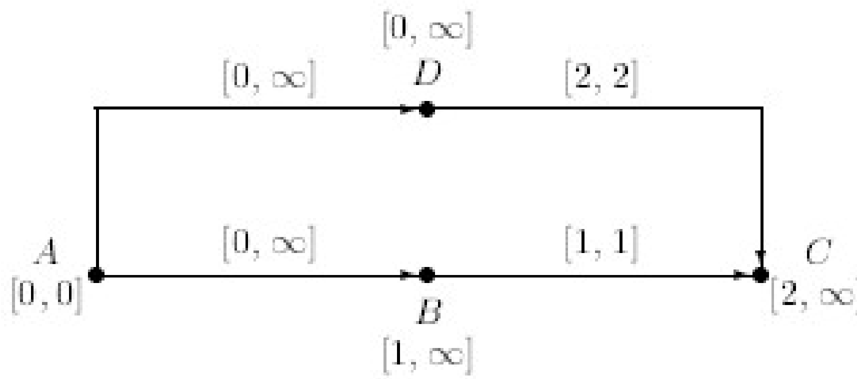
Remote Agent

- Solution: compile temporal constraints to an efficient network

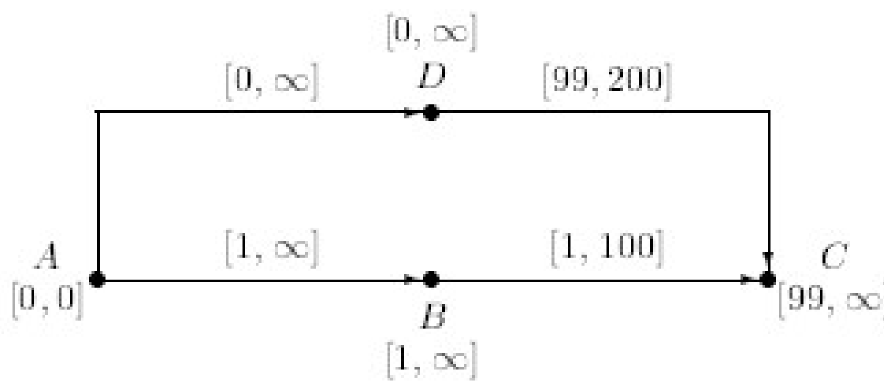


Remote Agent

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

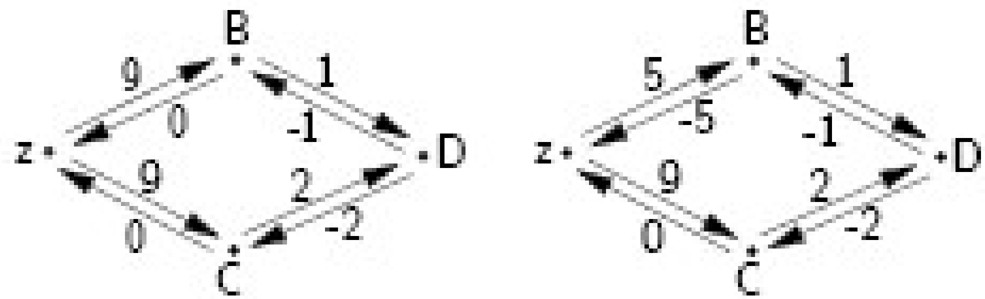


  - Occorre rendere la rete dispatchable aggiungendo vincoli impliciti (e.g. D prima di B)
  - Compilare la rete in forma dispatchable:
    - Introdotti vincoli impliciti
    - Tolti vincoli ridondanti



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

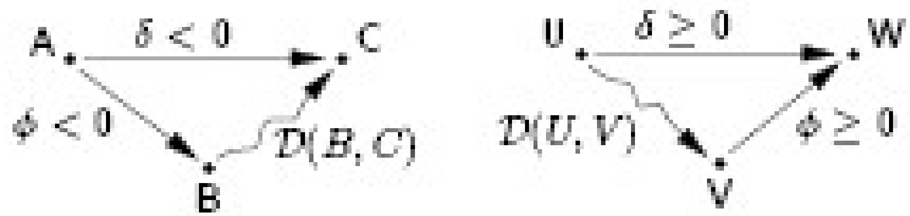
Dispatchability*

- 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

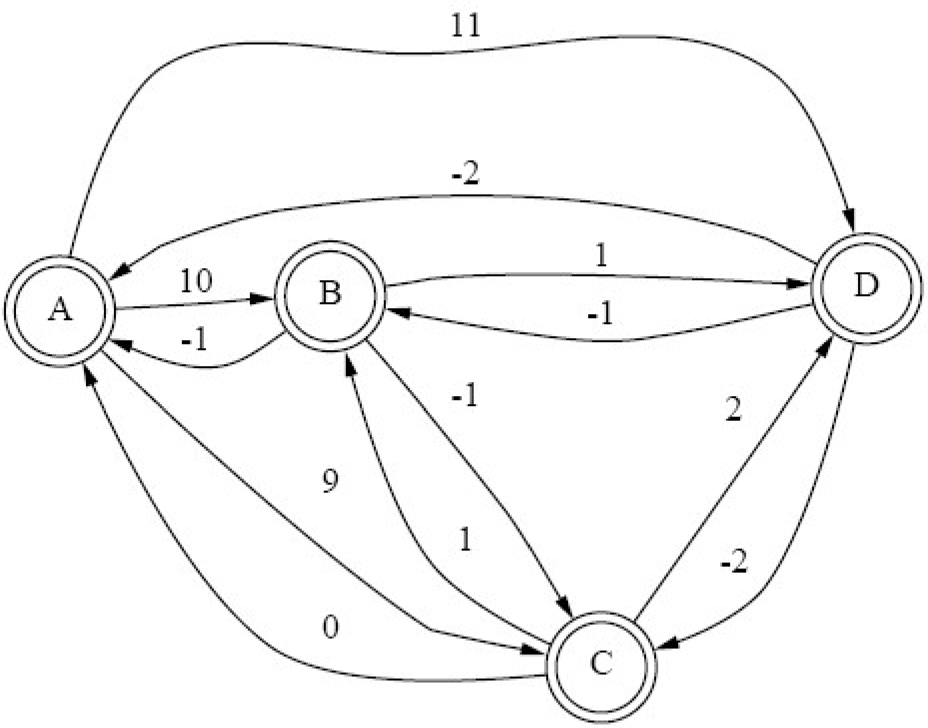
Lower and Upper Dominance*



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

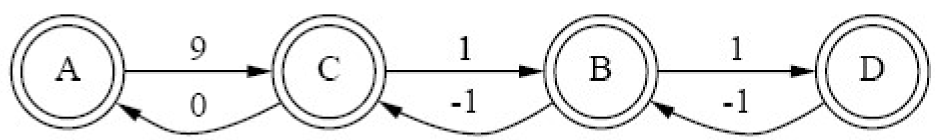
* (Muscettola, Morris, & Tsamardinos 1998)

Dispatchability



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

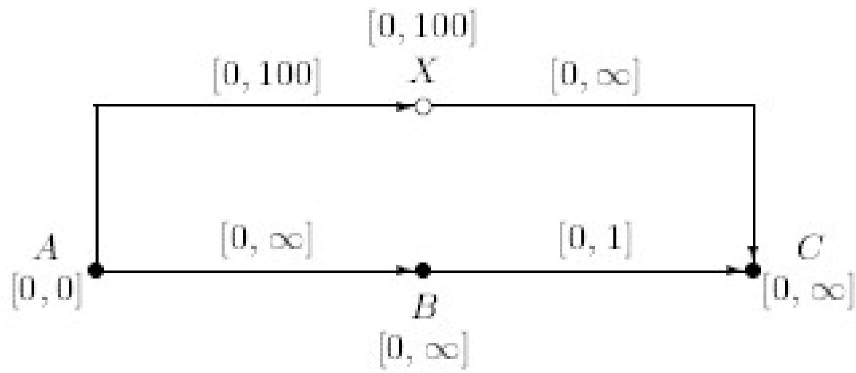
Controllability Issues*

- 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.

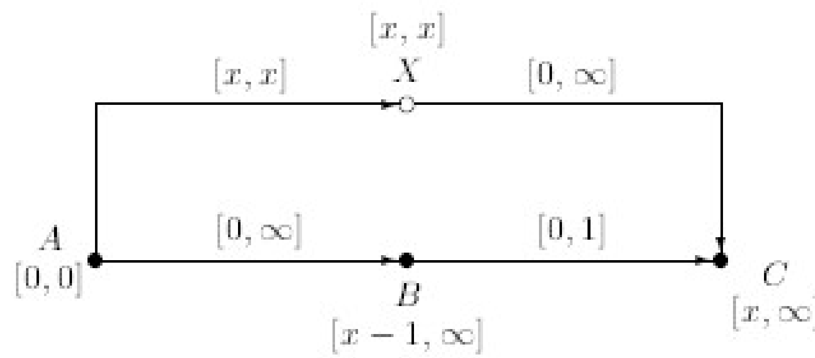
* (Vidal & Ghallab 1995; Vidal & Fargier)

Controllability

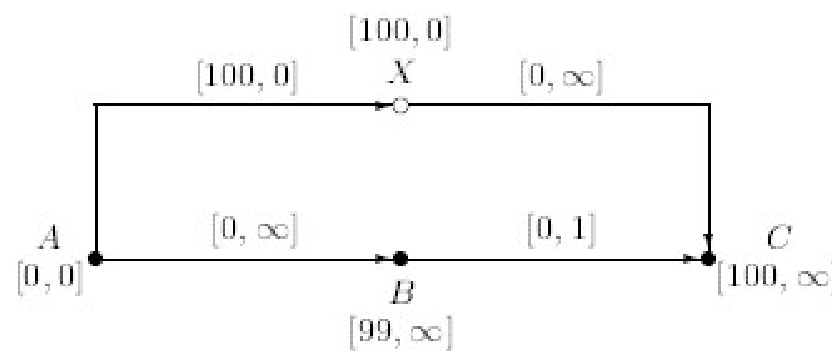
- Gestire eventi non controllabili
- Es. Se B schedulato prima di X, B vincola X



  - 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.

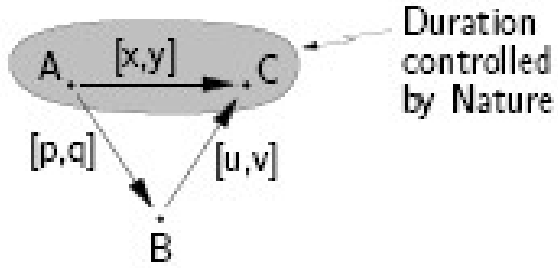
Controllability

Checking Dynamic Controllability*

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

- *Triangular Reductions*
- *Wait Propagation*

Triangular Reductions



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

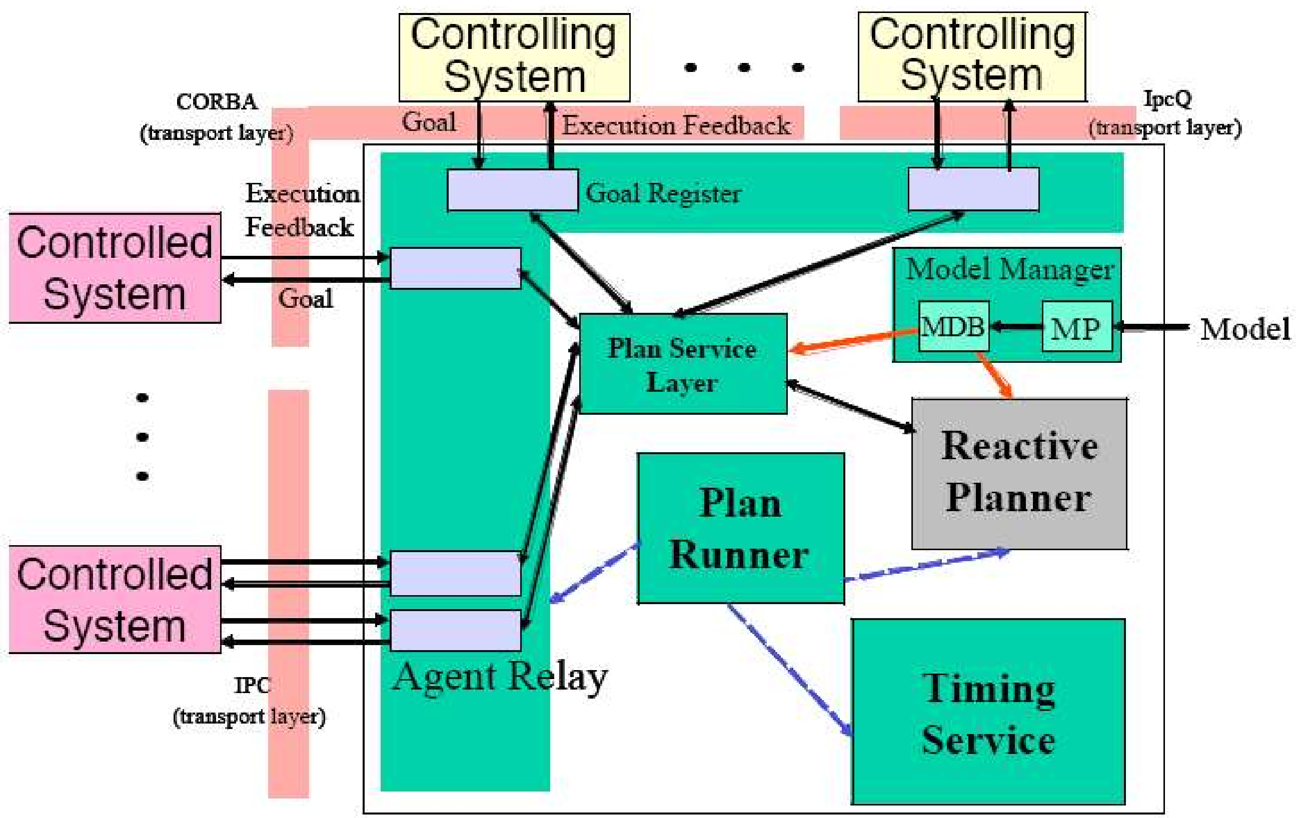
Triangular Reductions (ctd.)

- If u < 0 and v ≥ 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

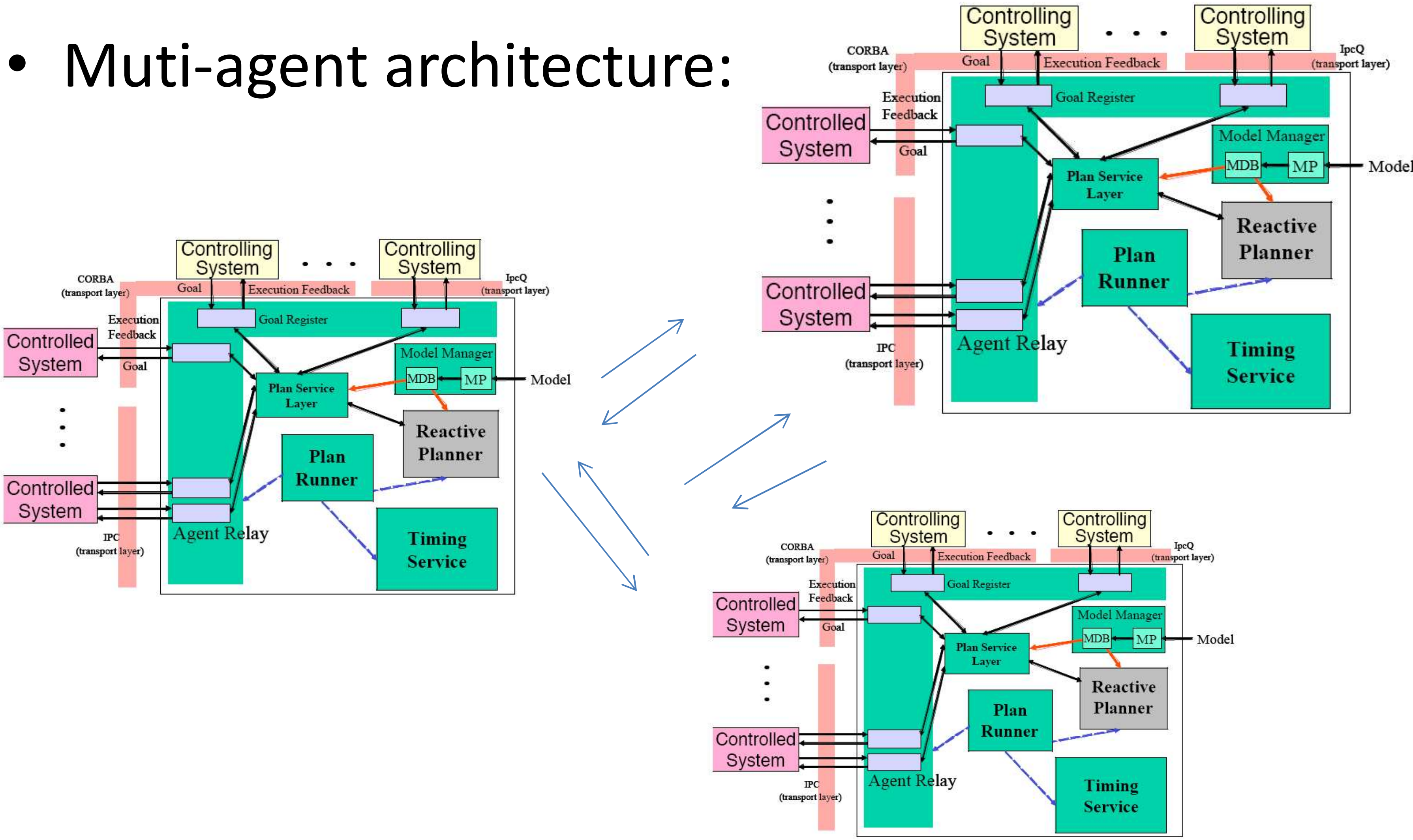
- Evoluzione del RA: reactive and deliberative planning



EUROPA Planner:
https://github.com/nasa/europa

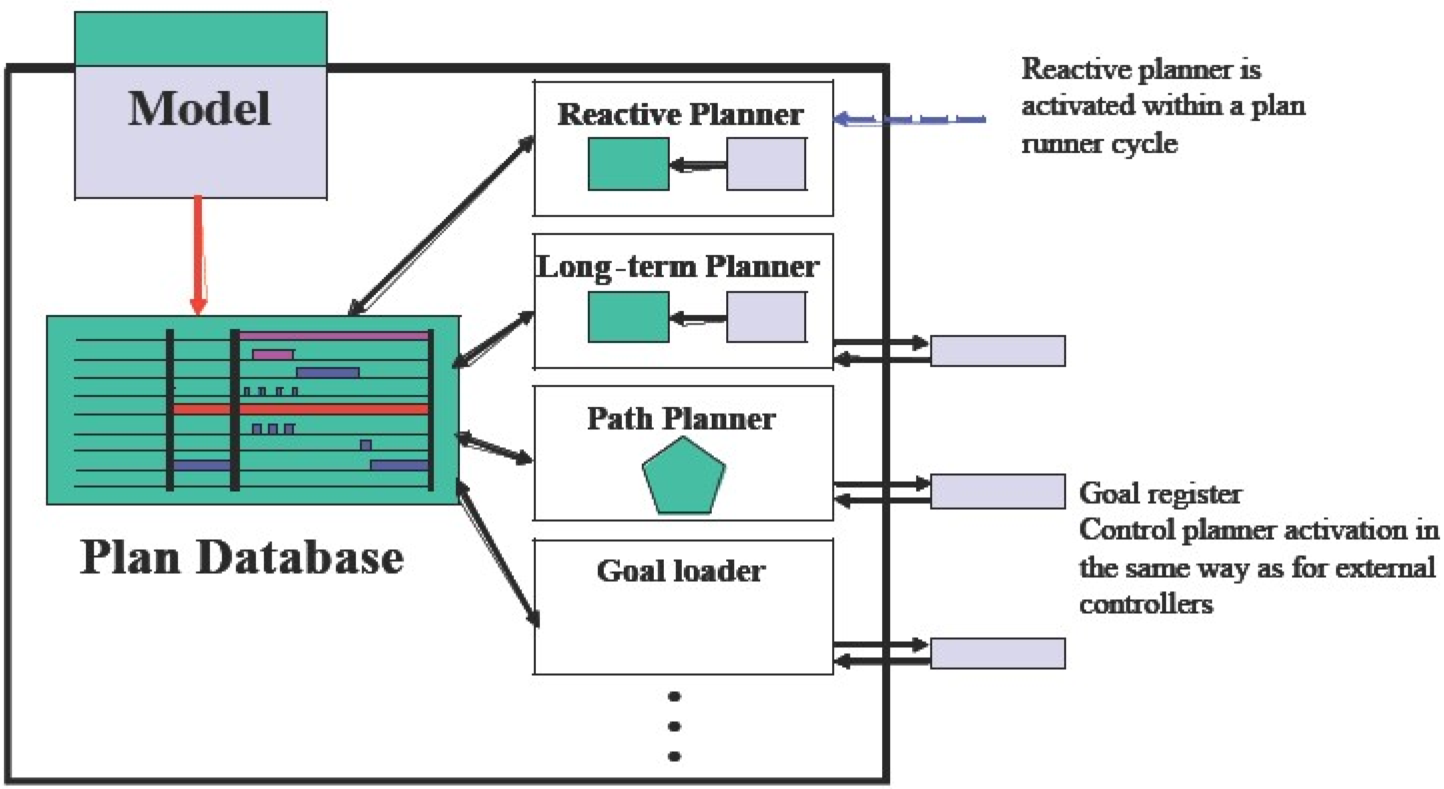
IDEA Architecture

- Muti-agent architecture:



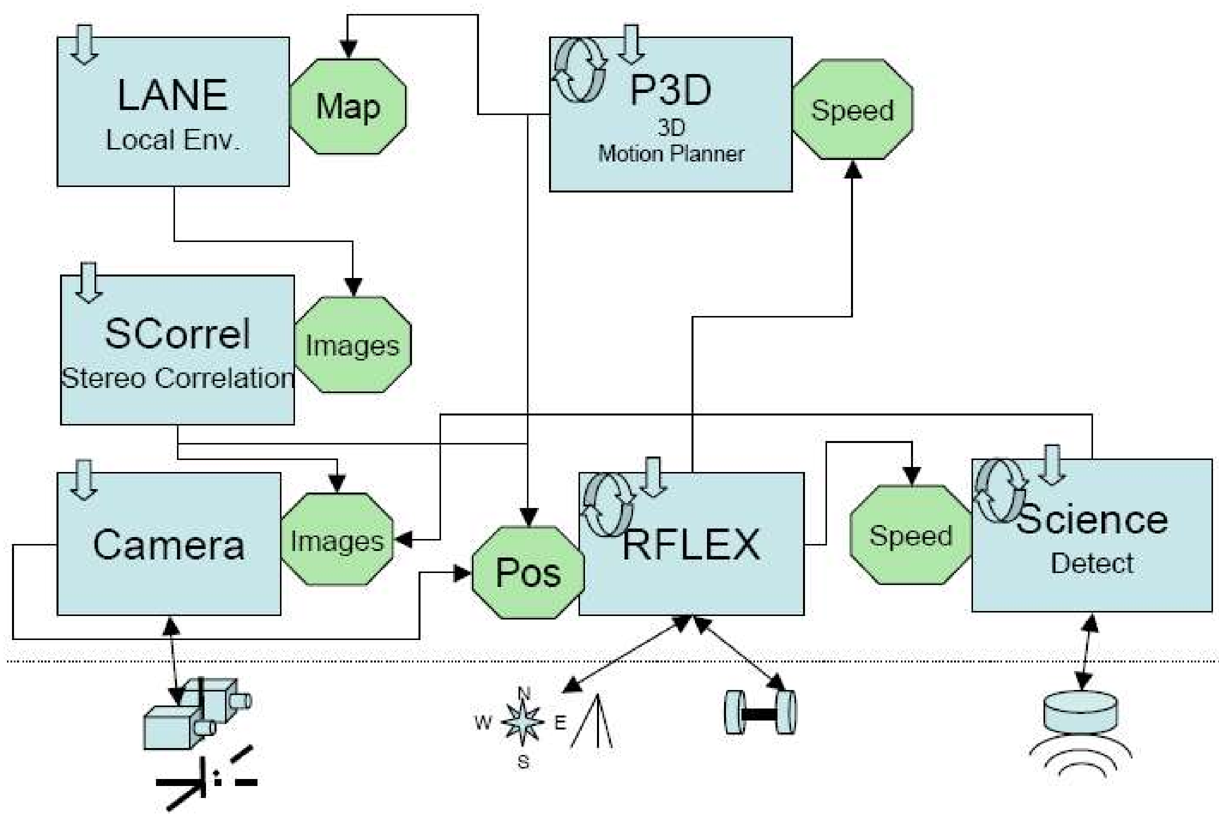
IDEA Architecture

- Reactive planning come controllo
- Interazione deliberative and reactive planning



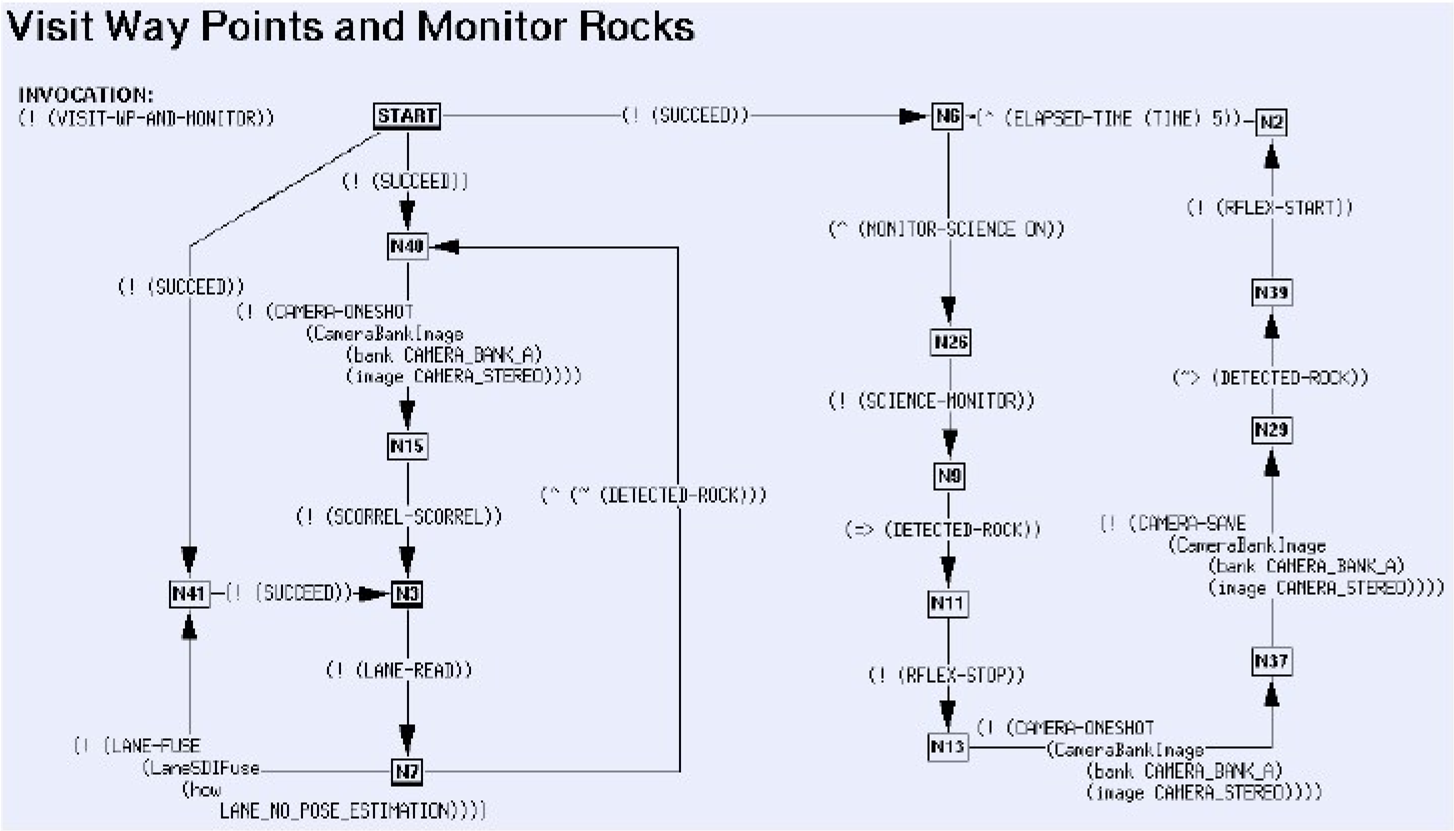
Functional Layer

- GenoM (LAAS)



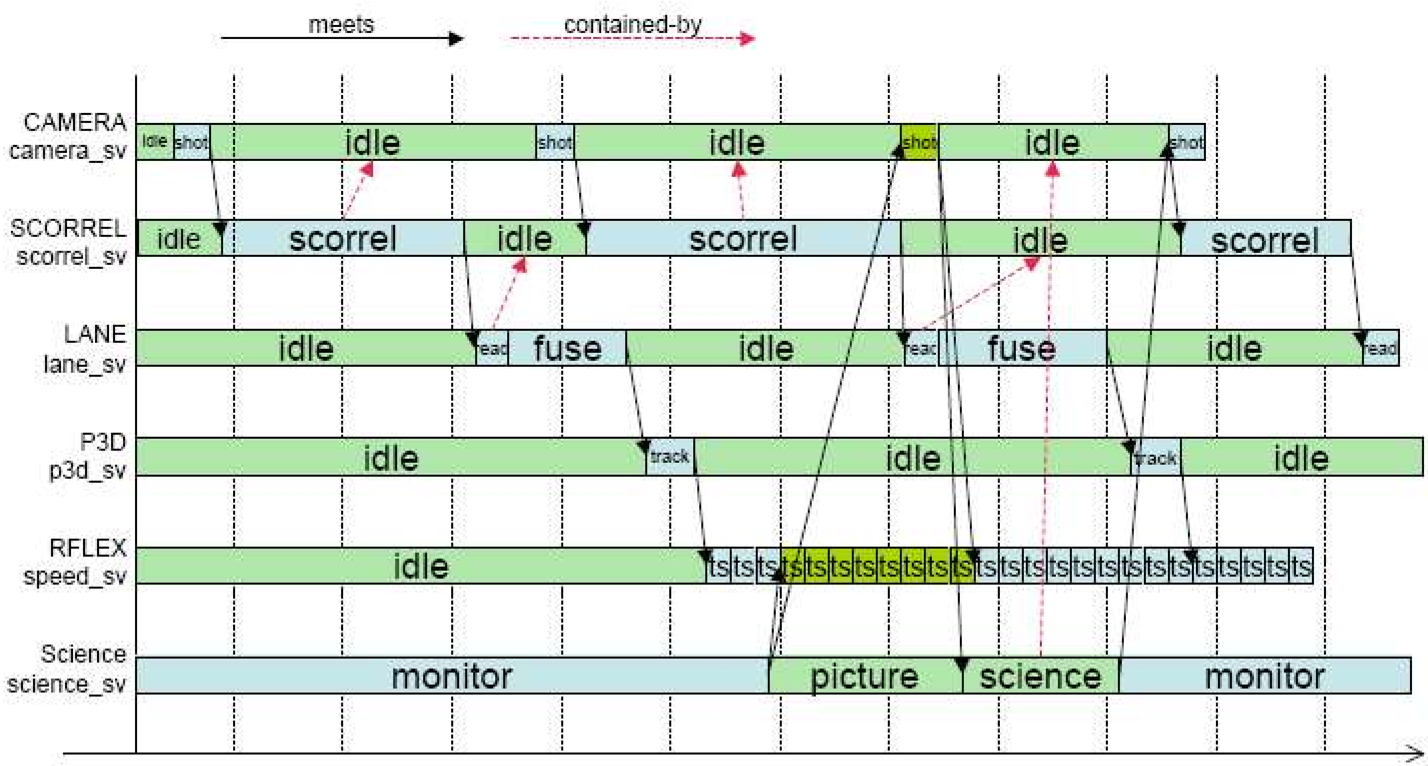
PRS Controller

- A procedural controller (vedi dopo …)



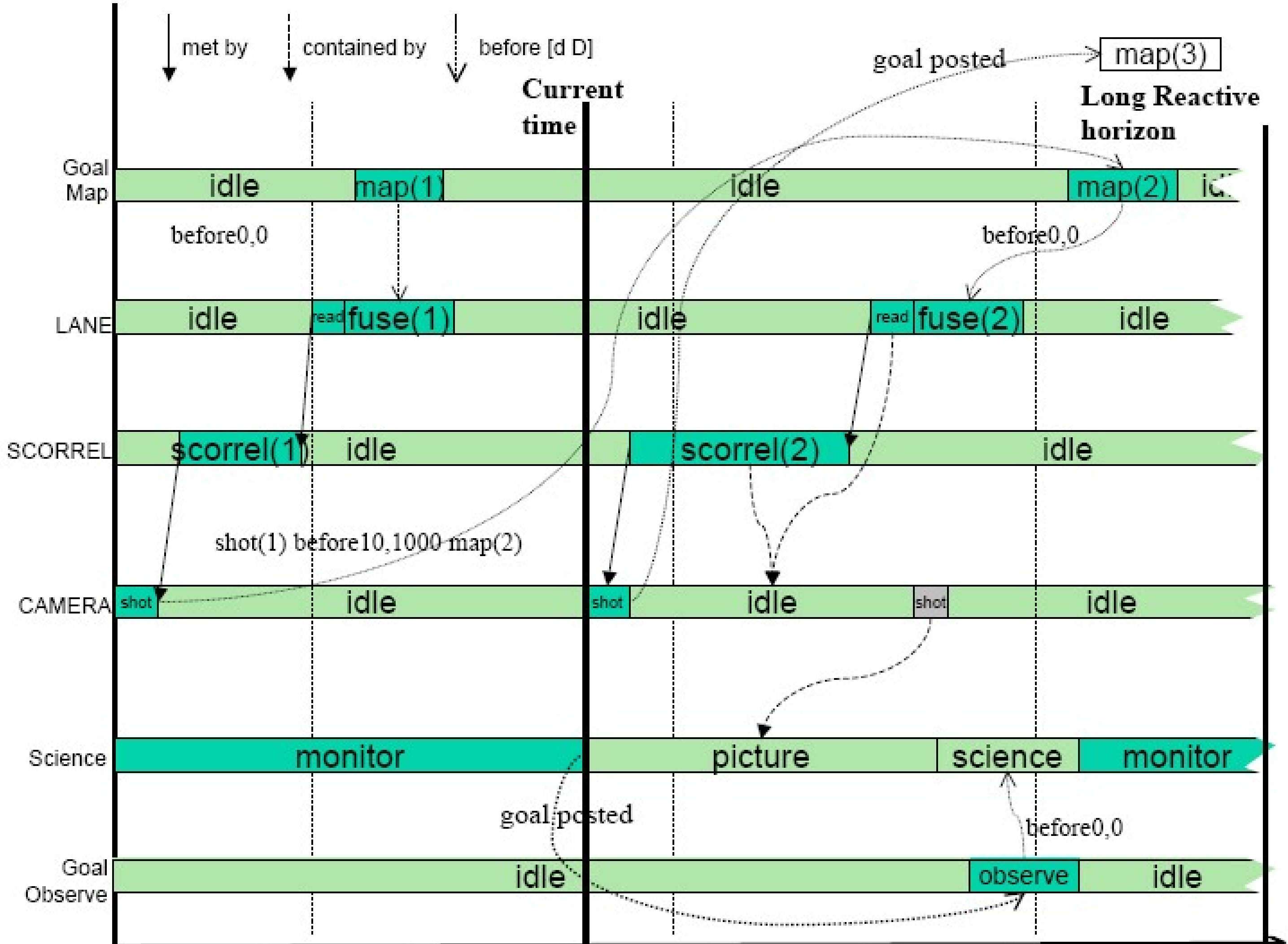
IDEA Architecture

- Attività pianificate e schedulate (plan database):



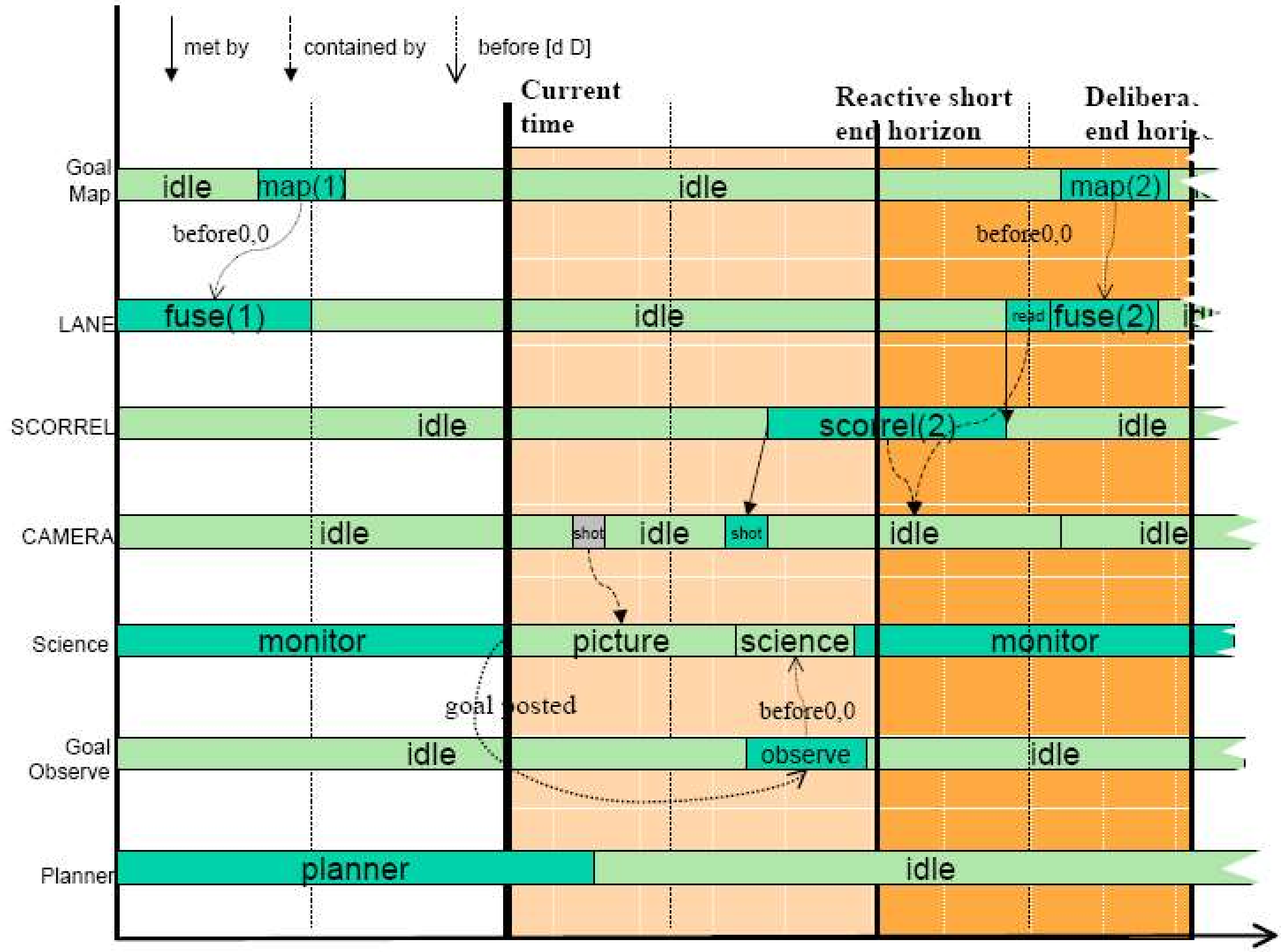
IDEA Architecture

- Reactive Planning



IDEA Architecture

- Reactive and Deliberative planning



Task Planners

- Classical planner
  - FF
    - https://fai.cs.uni-saarland.de/hoffmann/ff.html
- Hierarchical Planner
  - SHOP:
    - https://www.cs.umd.edu/projects/shop/
  - PANDA:
    - https://www.uniulm.de/en/in/ki/research/software/panda/
- Temporal Planner
  - EUROPA
    - https://github.com/nasa/europa
  - OPTIC
    - https://github.com/nasa/europa
- Collectors
  - ROSPLAN
    - https://kcl-planning.github.io/ROSPlan/documentation/

Planning Problems

- Path and Motion planning
  - Map/Configuration space
    - Methods: A*, PRM, RRT, etc.
    - Solution: path/trajectory
- Decision Theory Planning
  - Markovian Models
    - Methods: MDPs, POMDPs solvers (e.g., Dynamic Programming)
    - Solution: Policy
- Reinforcement Learning
  - Markovian Models
    - Methods: SARSA, Qlearning, etc.
    - Solution: Policy
- Task Planning (and Scheduling)
  - Planning Domains (symbolic representations)
    - Methods: Classical Planning, POP, HTNs, Temporal Planning, etc.
    - Solutions: sequence of actions, partial order plans, flexible plans, etc.