From Scheduling to Planning with Timelines: A history of successful applications in Space (Part 2)

Simone Fratini
European Space Agency
European Space Operations Center

ICAPS Summer School
04/06/2013

European Space Agency

Outline

1. Background
    a. Realistic Planning Problems
    b. Solving Problems with Planning and Scheduling

2. Theoretical Foundations:
    a. Basic Modelling Principles
    b. Search Space Representation
    c. Problem Solving with Timelines

3. Architectures
    a. Generic Architecture
    b. Real Examples of Architectures in use at ESA/NASA

BACKGROUND

Applications of AI Planning

Aerospace

Space

- Satellite, Launcher manufacturing
- Launch site operations
- Spacecraft operations
- Space users
- Autonomous space exploration

Aviation

- Manufacturers and suppliers
- Airlines, Airports
- Maintenance and repair organizations
- Air Traffic Control

P. Laborie– Planet School – Sept. 2002

Applications of AI Planning

Defense

Military logistic applications

Rescue and Military operations planning

Non-combatant Evacuation Operations

Emergency Assistance

Air Campaign Planning

Army Unit Operations

Coalition Operations

Applications of AI Planning

New information technology

Planning & Scheduling for the Web

Workflow, Business Process Management

Systems Management Aids

Help Desks and Assistants

Personal Assistants

Data analysis tasks, Database query

Virtual reality or other simulated environment

Computer games (bridge)

Applications of AI Planning

Industry

Process planning

Factory automation

- Production line scheduling
- Assembly planning
- Maintenance planning

Electronic design and manufacturing

Logistics

Project planning

Construction planning

Engineering Tasks

Management Tasks

Applications of AI Planning

Civil Security and Environment

Crisis Management

Evacuation Planning

Search & Rescue Coordination

Oil Spill Response

Medical Applications

Bioinformatics

Personal Active Medical Assistant

"Realistic" Planning Problems

1. Time
2. Numerical Reasoning
3. Concurrent actions
4. Context–Dependent Effects
5. Interaction with Users, "man in the loop"
6. Oversubscribed Problems
7. Focus on Optimization rather than on Achievement
8. Robustness, Flexibility
9. Execution Monitoring
10. Re-Planning & Re-Scheduling
11. Scalability
12. Integration of external solving processes
13. Need for Formal Verification and Validation of Plans and Schedules

“Realistic” Planning Problems

1. knowledge about the domain
   - Timing, causal relationships, resources
2. knowledge about “good plans”
   - Oversubscription, Optimization, Robustness, Flexibility
3. explicit search-control knowledge
   - Domain specific know-how, Optimization
4. knowledge about interacting with the user and about user’s preferences
   - “man in the loop”
5. knowledge about plan repair during execution
   - Execution monitoring, re-planning and re-scheduling

Six Challenges (D. Smith, Planet '03)

1. Exogenous conditions & events
    a. Communication windows
    b. Observation windows
    c. Lighting conditions
    d. Orbit insertion
    e. Celestial events

2. Over-subscription problems
    a. Many observations/experiments
    b. Prioritized
    c. Limited time, energy, data storage, cryogen…

3. Uncertainty

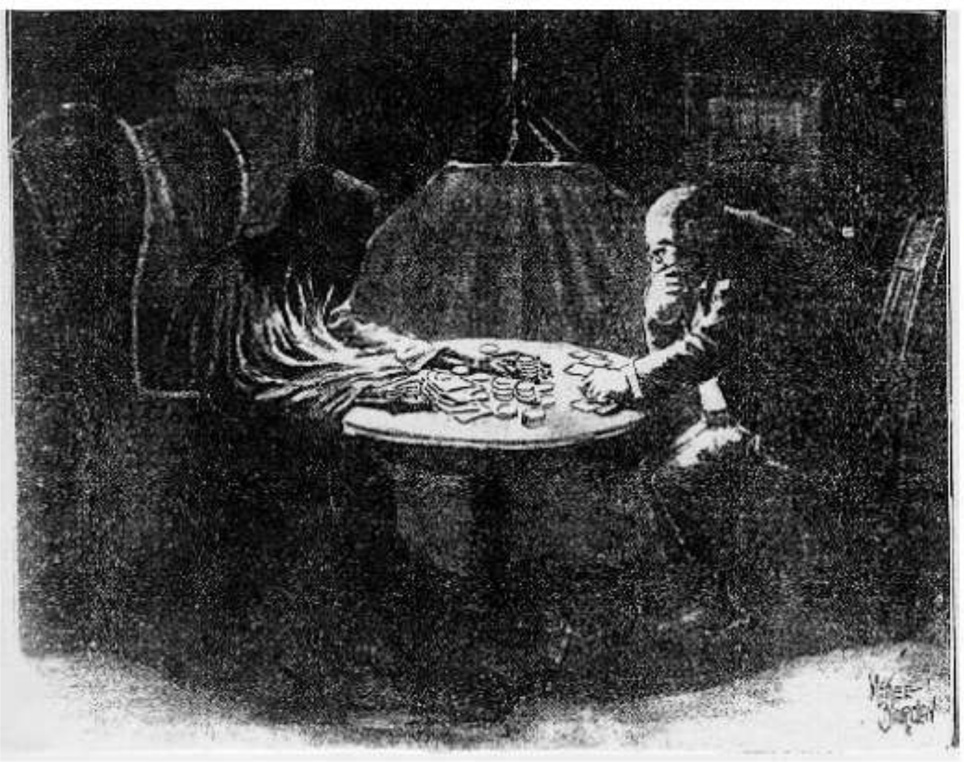
Six Challenges (D. Smith, Planet '03)

4. Ramifications
    a. Physical model
    b. Switches/valves

5. Plan Revision
    a. Rolling time horizon
    b. New goals
    c. Unexpected state

6. External Reasoning
    a. Flight Dynamics
    b. Thermal Model

What is the opponent?



Do not underestimate him

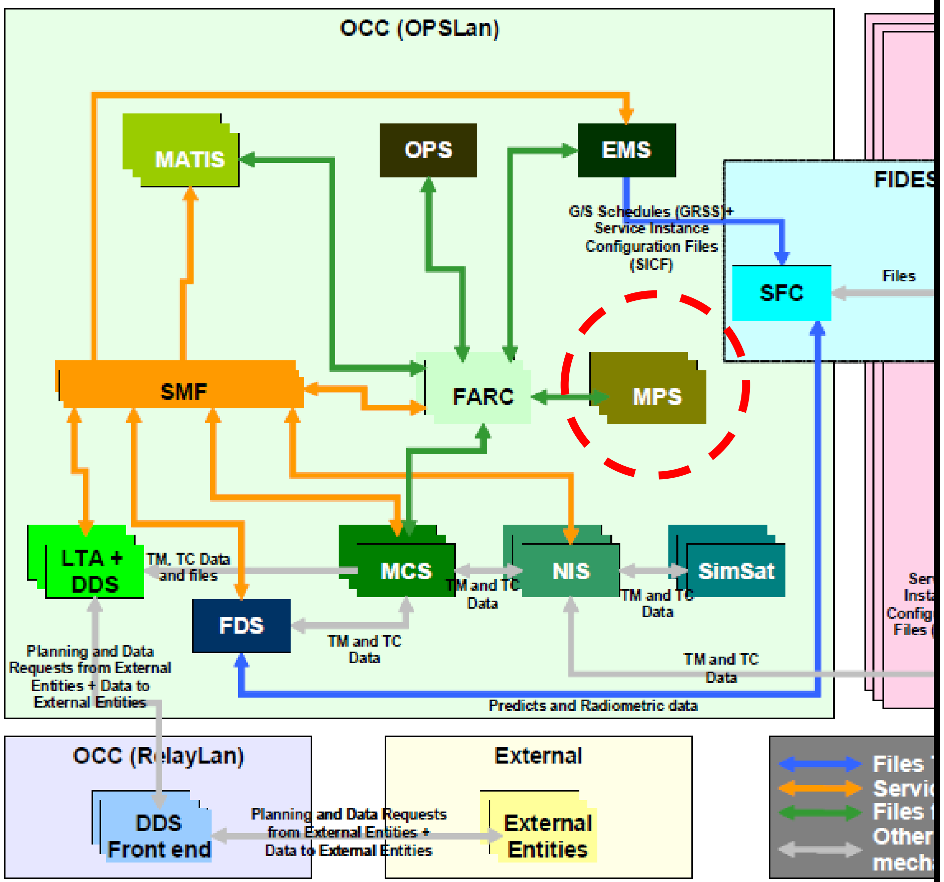
Do not expect him to show you any mercy when you make mistakes

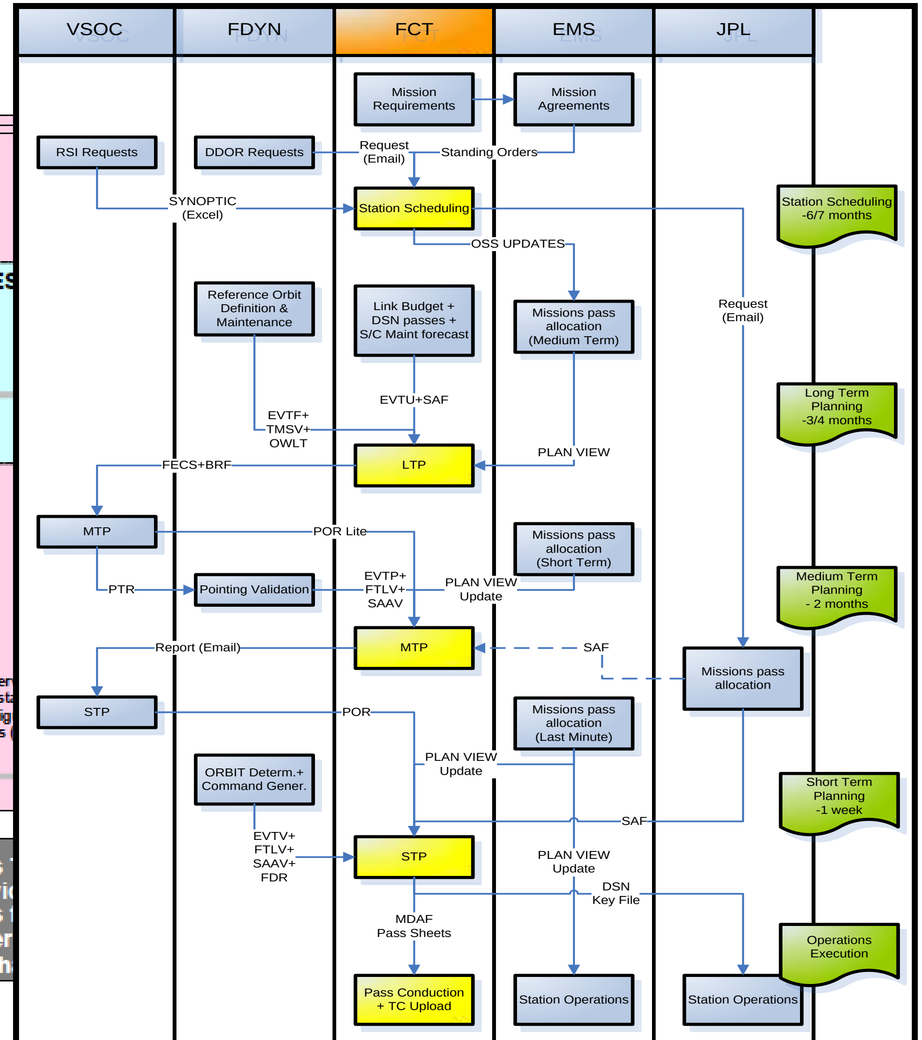
Always remember that space, even more than air and sea is the most unforgiving environment in which to play him

[Courtesy of D. Evans, ESA. “The ladybird guide to spacecraft operations”. Introductory course at ESA for Spacecraft Engineers]

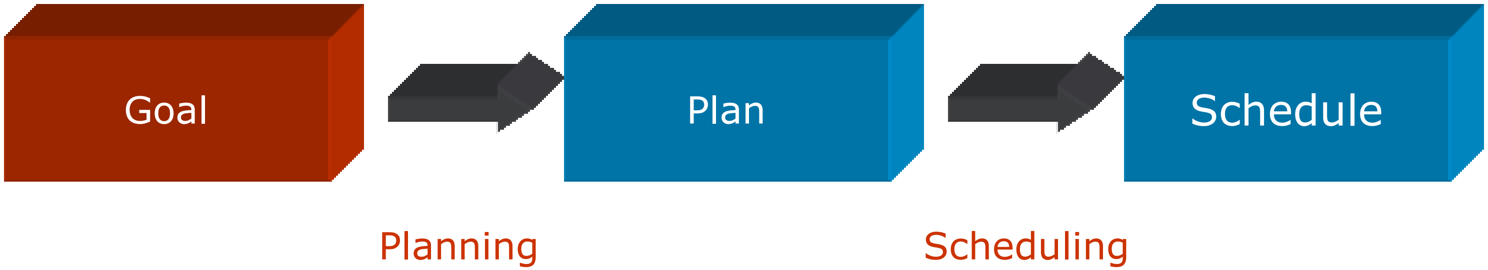
Ideal Scenario...

A Real Scenario…





Planning and Scheduling (in theory)



Planning and Scheduling (in theory)

Goal → Plan → Schedule

Planning

Scheduling

Planning and Scheduling (in theory)

Goal → Plan → Schedule

Planning

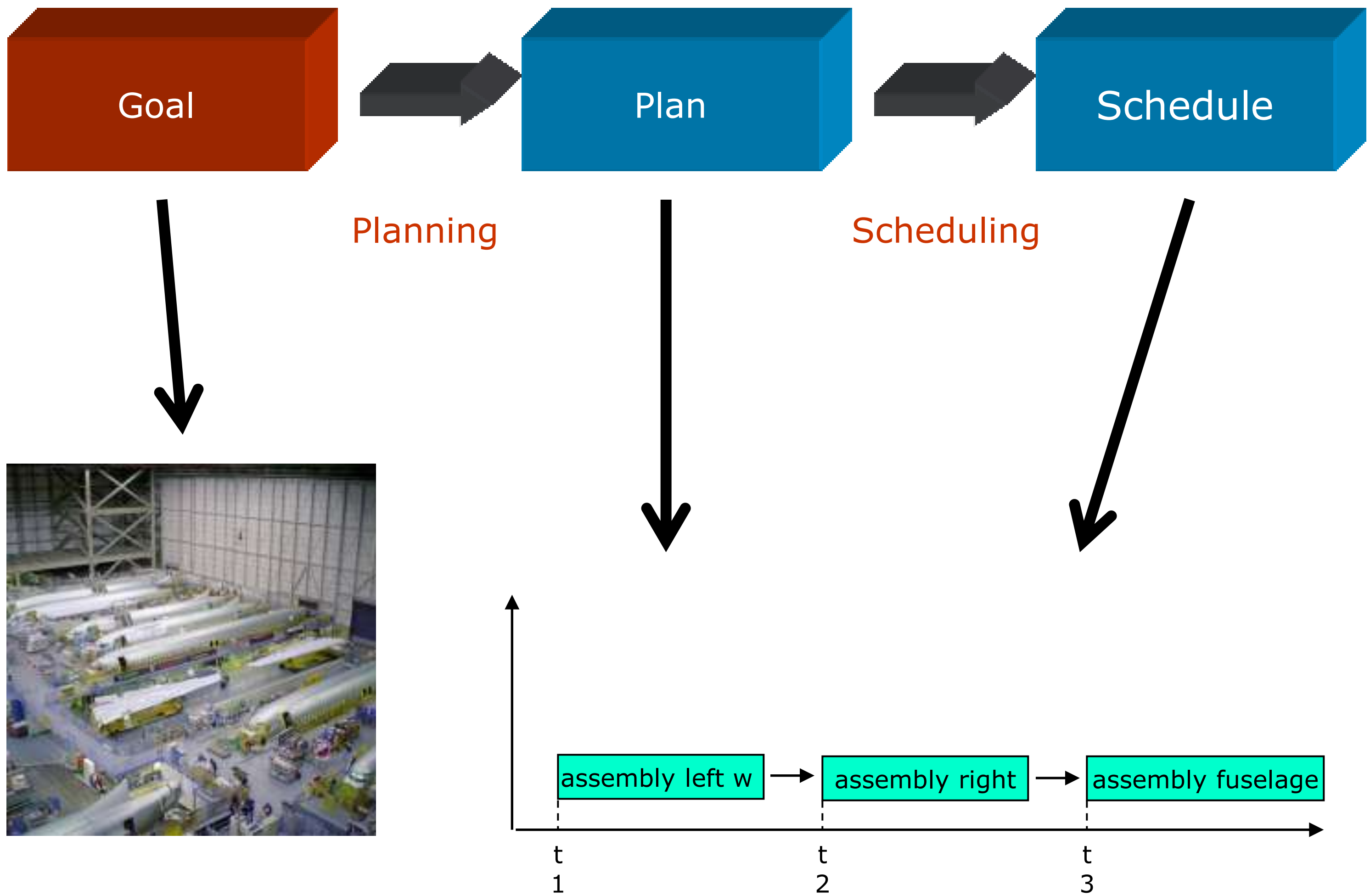
Scheduling

assembly left wing

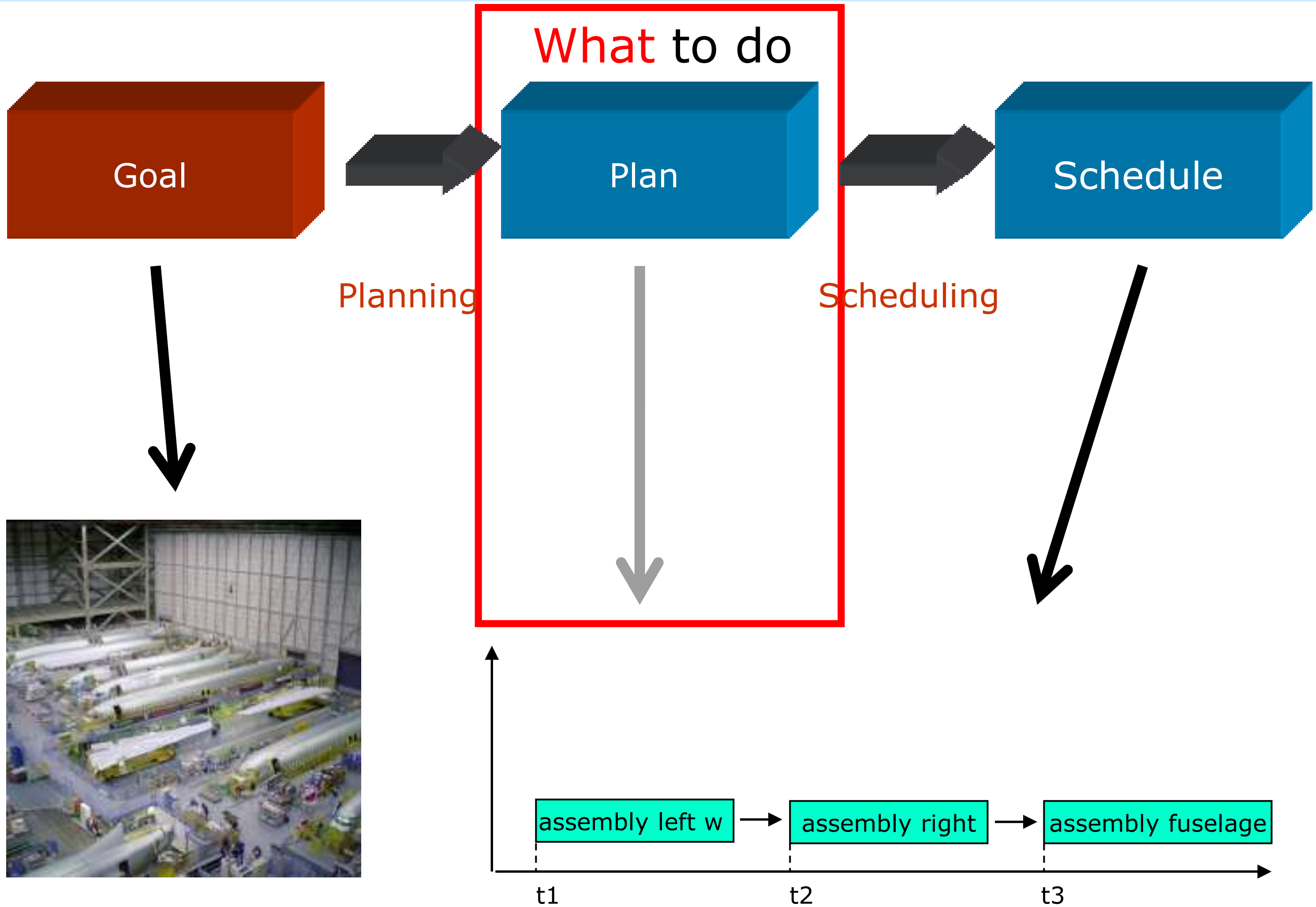
assembly right wing

assembly fuselage

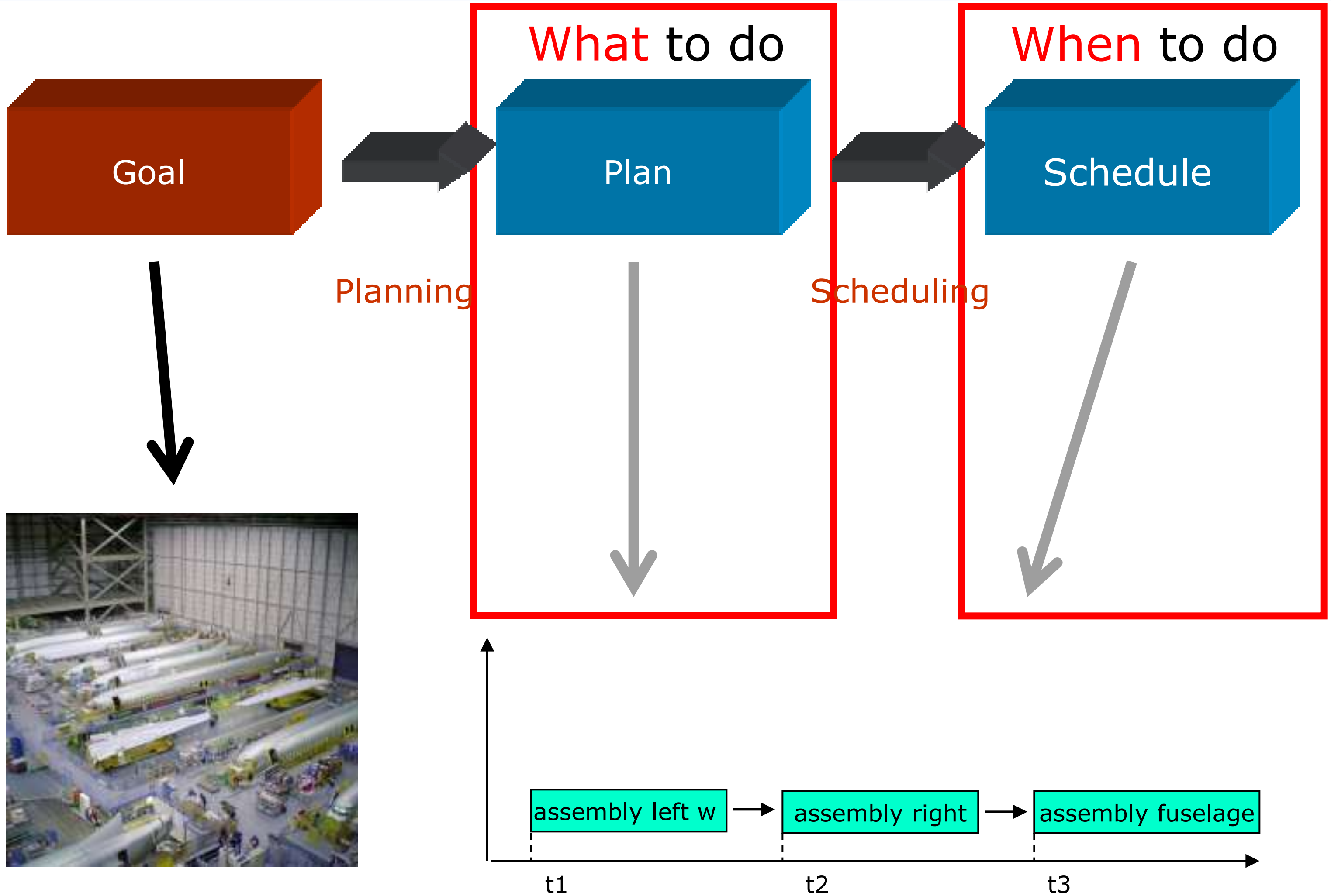
Planning and Scheduling (in theory)



Planning and Scheduling (in theory)

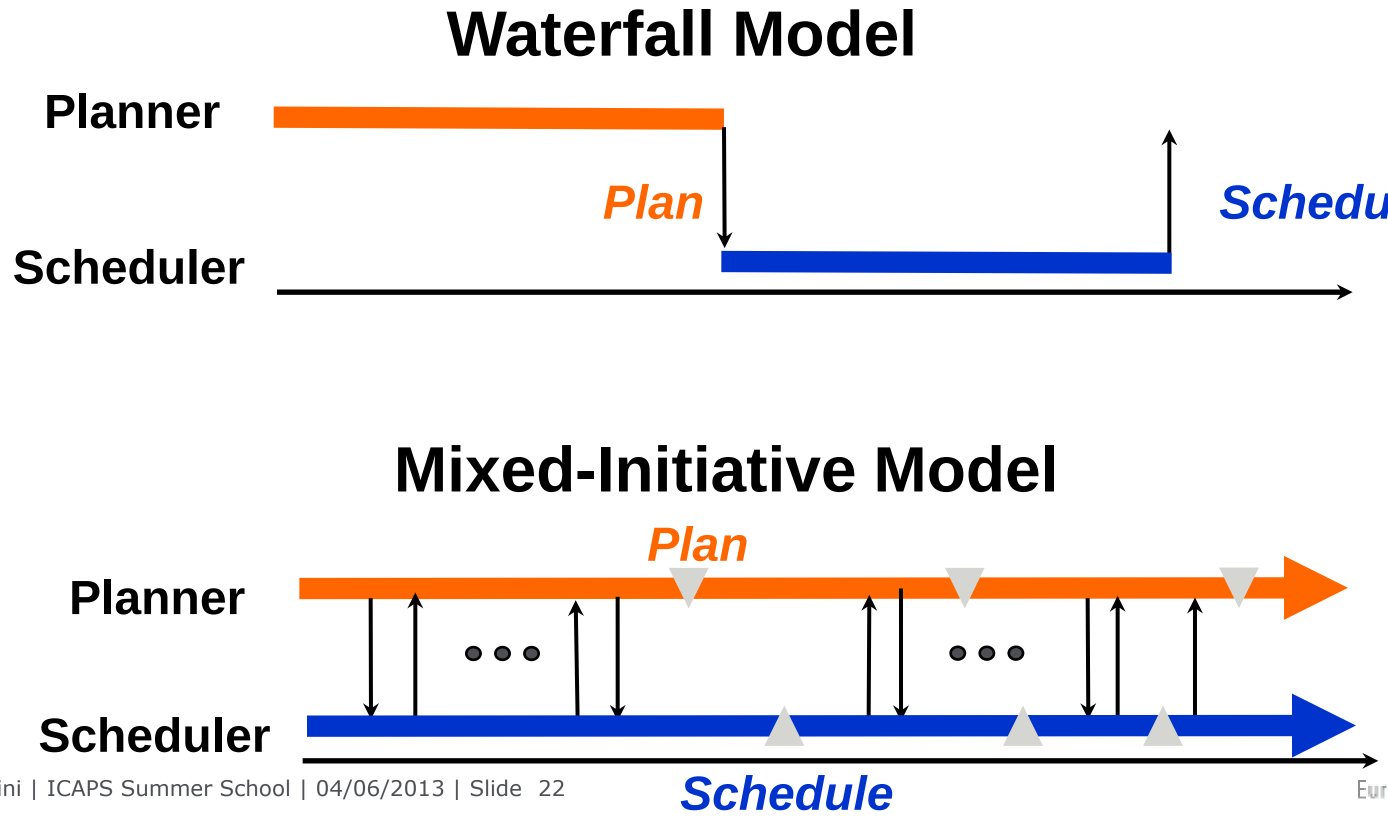


Planning and Scheduling (in theory)



Integrating Planning and Scheduling

"Planning & scheduling are rarely separable"



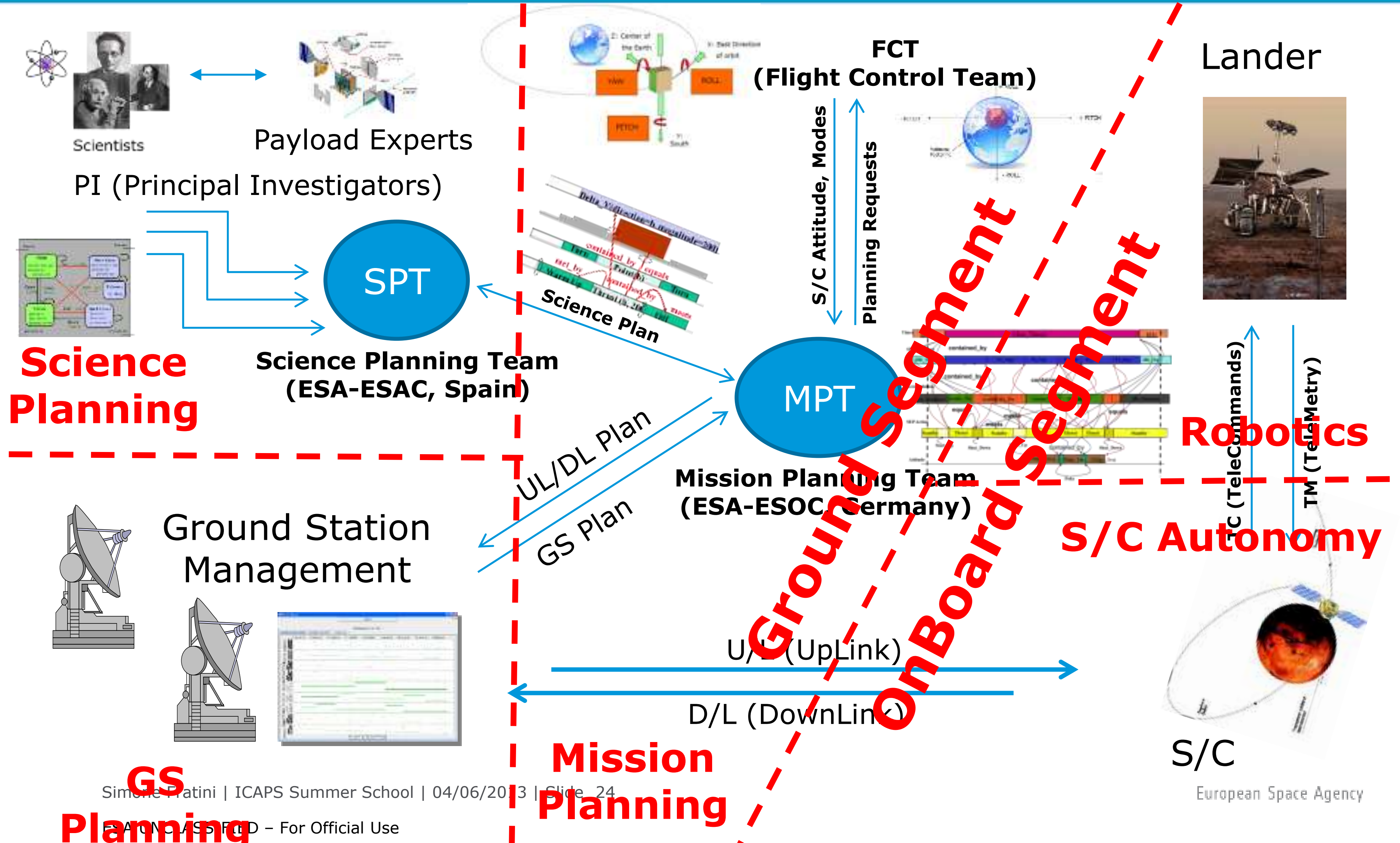
Planning and Scheduling in space

1. Integrated Planning and Scheduling:
    a. Activities logically depend on when are scheduled and on which resources are available
    b. Activity schedule depends on logical constraints

2. Functional Scope:
    a. Ground Segment P&S
    b. On-Board P&S
    c. Lander/Robotics P&S

3. Temporal Scope:
    a. Long Term Planning (LTP) – Up to 6 month
    b. Medium Term Planning (MTP) – ~4 Weeks
    c. Short Term Planning (STP) – Weekly slices

Functional Scope



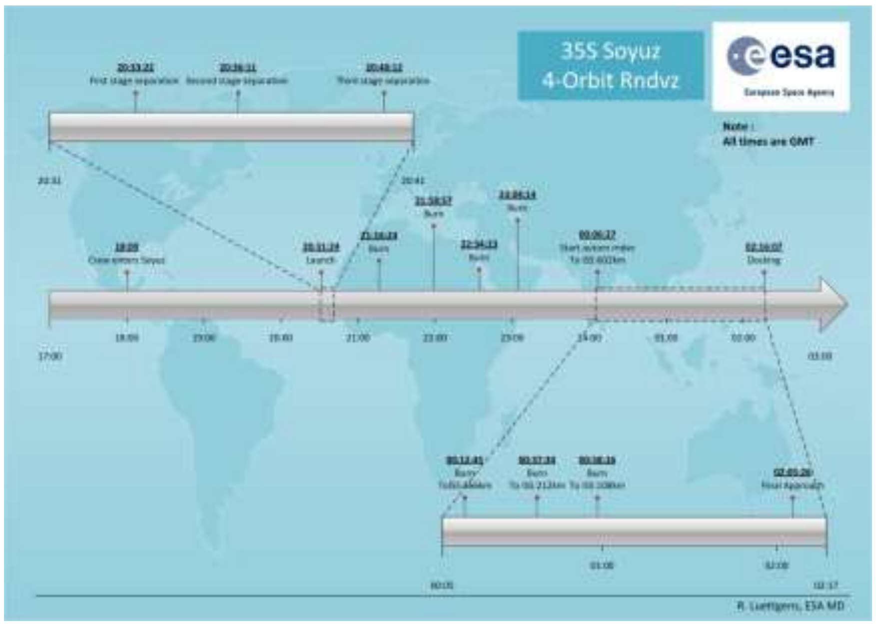
European Space Agency

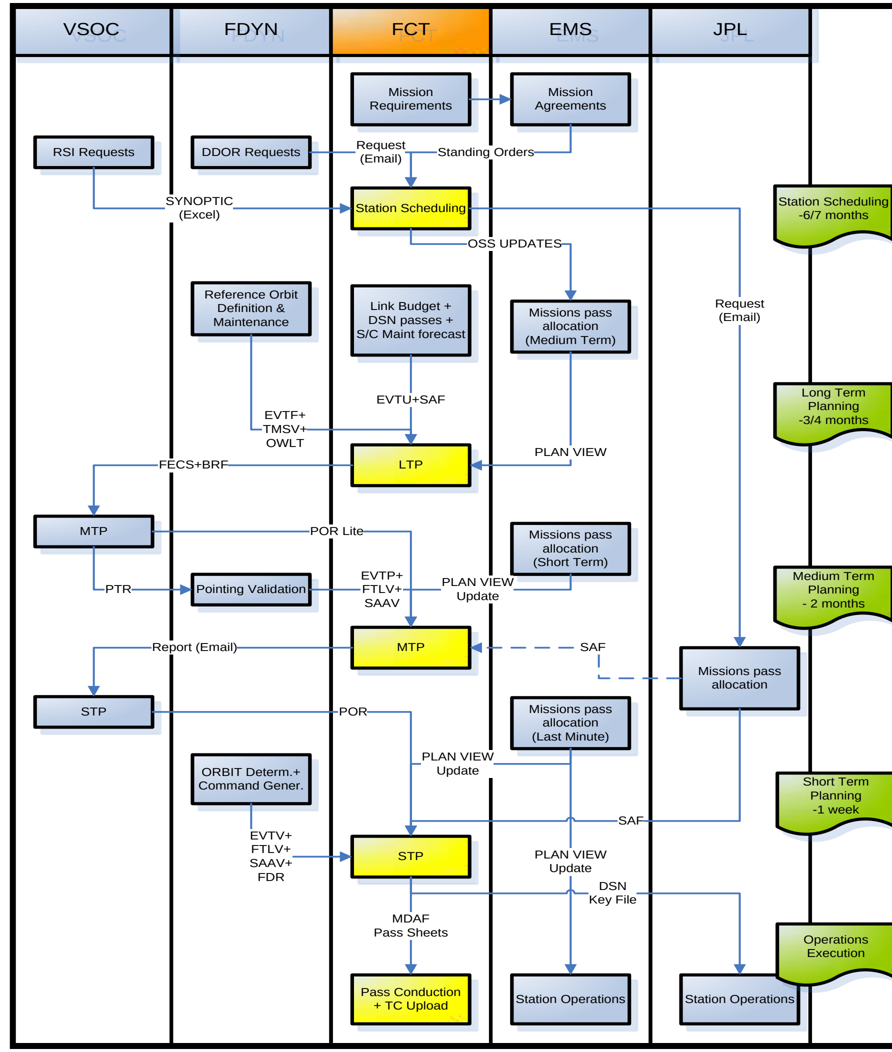
On-Board versus Ground Segment

1. Planning & Scheduling for on-board operations:
    a. Autonomy, “Closed-Loop” Control,
    b. on-board Plan & Execution, On-line assessment and revision
    c. Enable the opportunity to specify high-level science requests
    d. Plans are generated
    e. Verification and validation against on-board status

2. Planning & Scheduling for the ground segment:
    a. Role as decision support systems
    b. Interactive decision making
    c. User in control
    d. Subdivision of work between automated solvers (difficult and repetitive tasks) and human operator (strategic decisions, soft constraints)

Temporal Scope





A bit of history...

1. On Ground Segment
    a. HSTS (since 1990) – NASA
    b. MAPGEN (2003) – NASA/JPL
    c. Mexar 2(2005), Raxem (2006) - ESA
    d. APSI Framework (2007), MrSpock, AIMS, XMAS - ESA

2. On Board Segment
    a. Remote Agent (1999) - NASA
    b. ASE, EO-1 (2003) – NASA/JPL
    c. GOAC – Goal Oriented Autonomous Controller (2011) - ESA

THEORETICAL FOUNDATIONS

Outline

1. Basic Modelling Principles:
    a. Timelines, State Variables, Resources
    b. Domain Theory
    c. Problems and Solutions

2. Search Space Representation:
    a. Envelopes
    b. Types of timelines

3. Problem Solving with Timelines:
    a. Timeline Extraction Procedures
    b. Flaws Collection
    c. Flaws Resolution

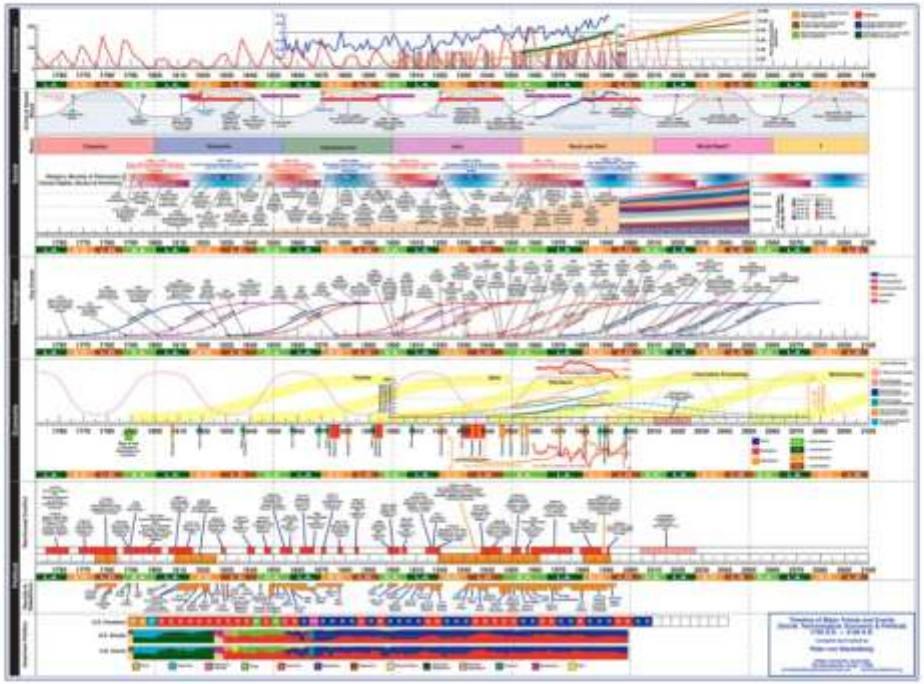
Basic Modelling Principles:

- Timelines, State Variables, Resources
- Domain Theory
- Problems and Solutions

Planning as...

Multiple Definitions:

- **Timeline-Based Planning**
- Planning with State Variables
- Constraint-based Attribute and Interval Planning (CAIP or C-BIP)
- Component-Based Planning
- ...



Pillars

1. Control Theory

[Passino, K. M. and Antsaklis, P. J. (1989). A system and control theoretic perspective on artificial intelligence planning systems. *Journal of Applied Artificial Intelligence*, 3:1–32.]

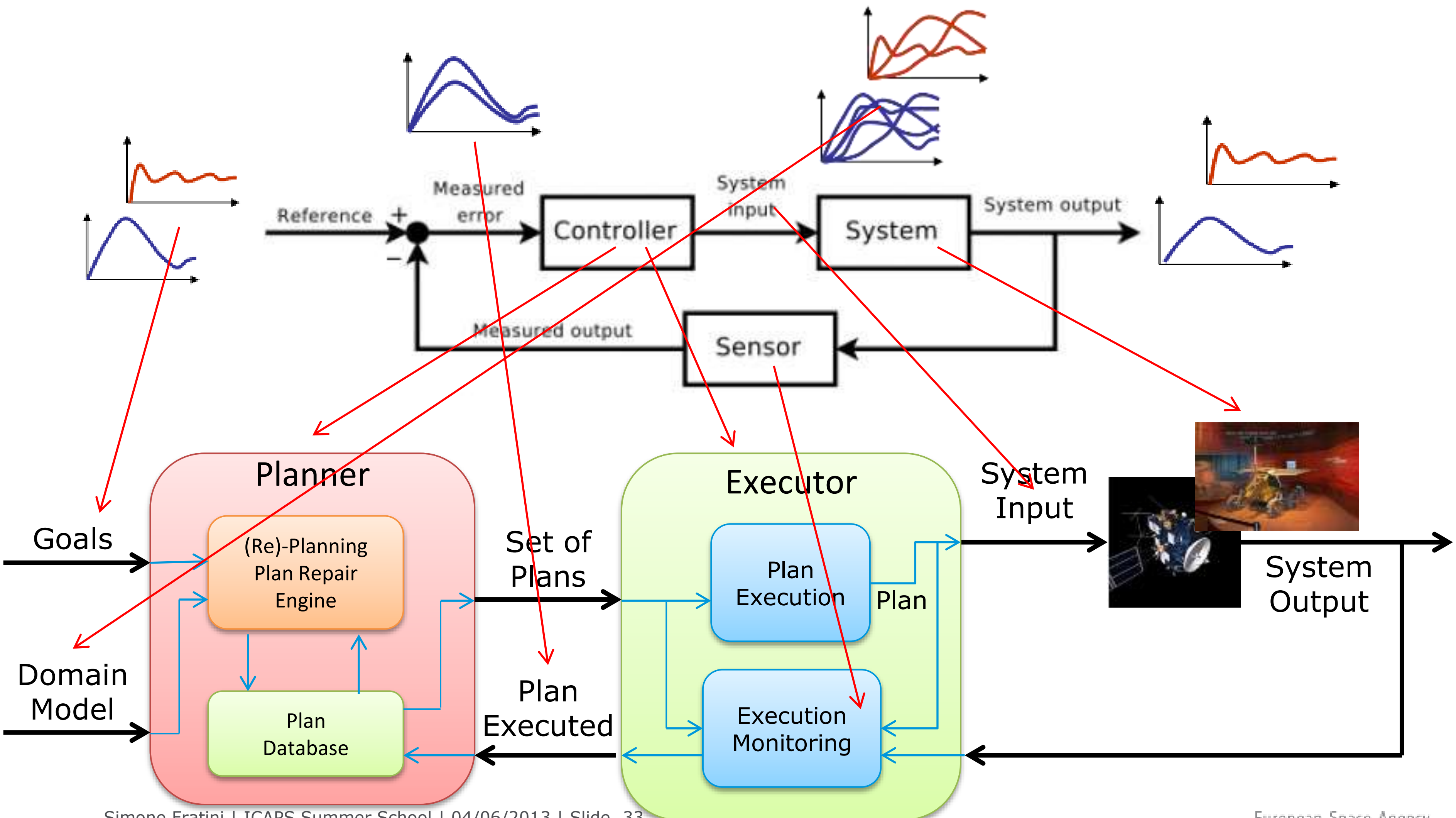
2. Constraint Satisfaction Problem

[Montanari, U. (1974). Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, (7):95–132.]

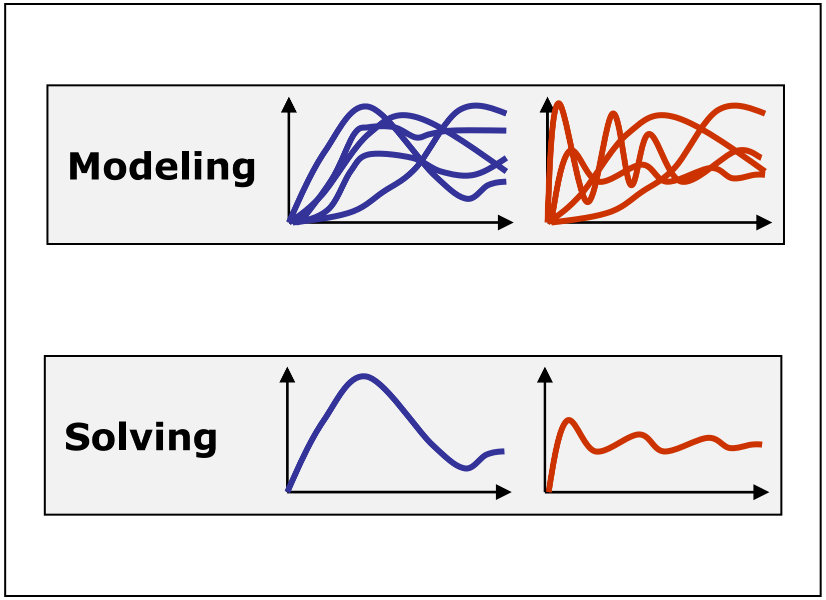
3. PCP Scheduling

[Muscettola, N. (1994a). HSTS: Integrating Planning and Scheduling. In Zweben, M. and Fox, M.S., editor, *Intelligent Scheduling*. Morgan Kauffmann]

Analogy with Control Theory



Timeline Based Approach

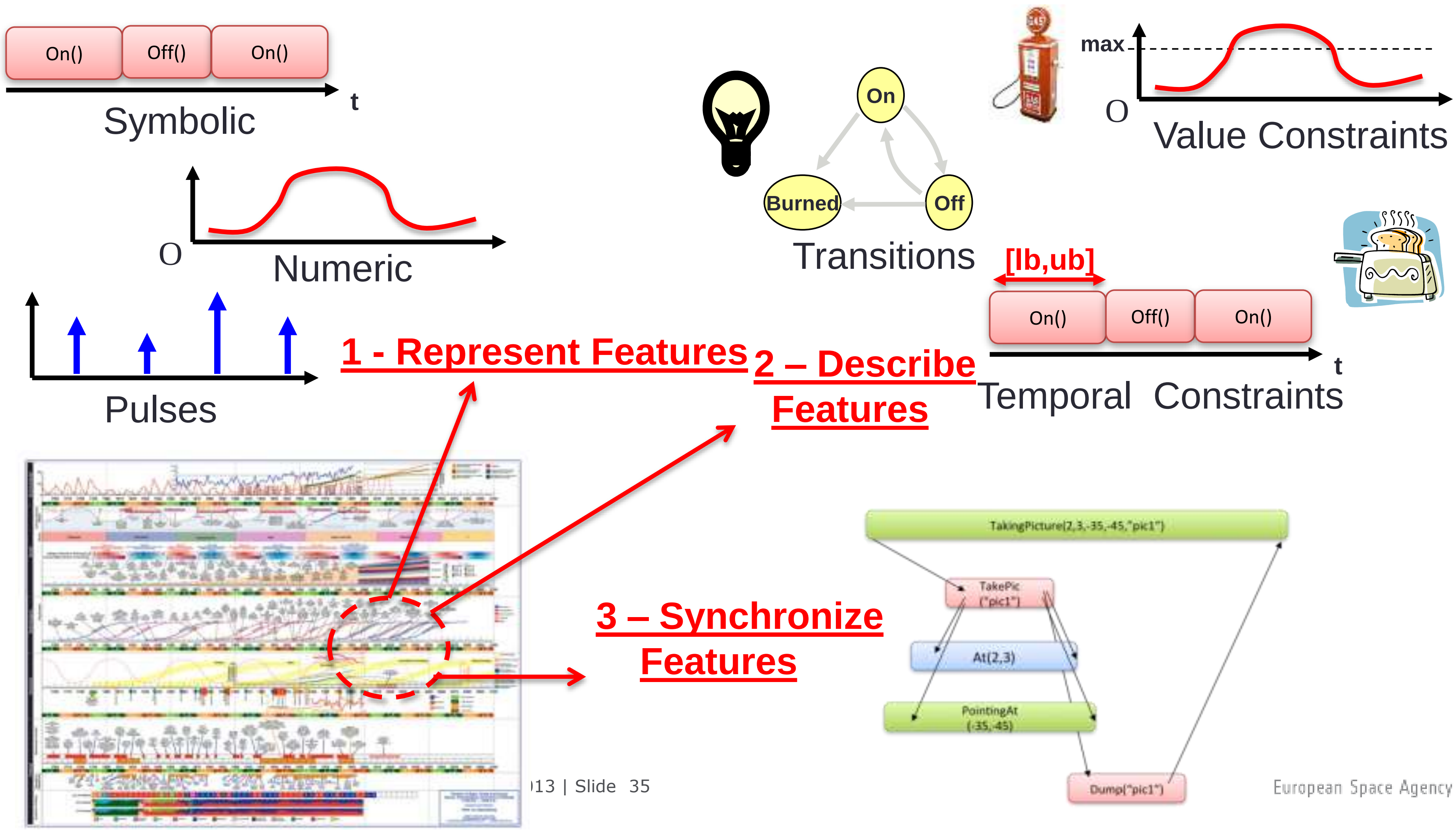


- Modelling
  - Focus on key features
  - Describe their possible consistent temporal behaviours
  - Represent the relevant constraints (domain theory)

- Solving
  - Synthesize timelines according to current goals satisfying modelled constraints

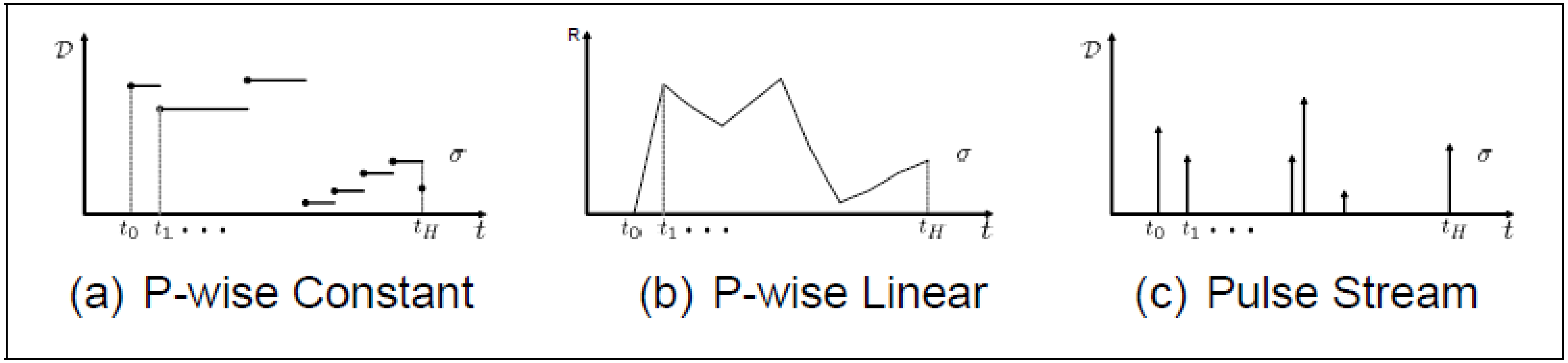
Problem Solving = Timeline synthesis

Timeline-Based Planning



Modelling Assumptions

1. The world is modeled as a set of **entities** (physical subsystems)…
2. whose properties can vary in **time**…
3. according with an **internal** logic…
4. or as a consequence of **external** inputs.



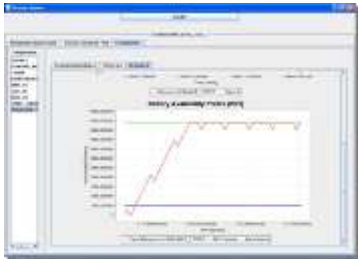
Components Profiles

1. The world is modeled as a set of **entities** (physical subsystems)…
2. whose properties can vary in **time**…
3. according with an **internal** logic…
4. or as a consequence of **external** inputs.

2 classes of sub-systems:

1. Symbolic
   (to represent operational modes)

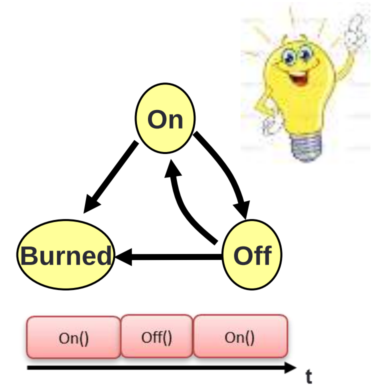
2. Numeric
   (to represent numerical variables)

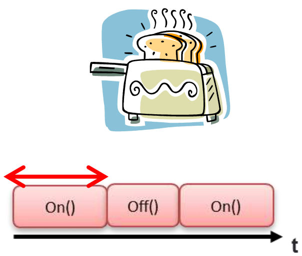


Profile Constraints

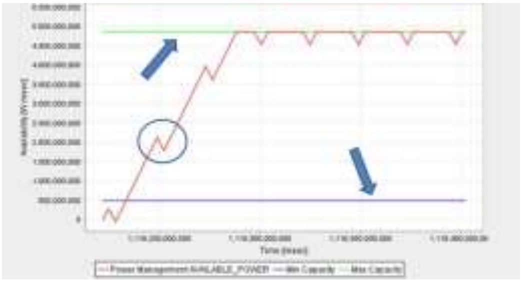
1. The world is modeled as a set of **entities** (physical subsystems)…
2. whose properties can vary in **time**…
3. according with an **internal** logic…
4. or as a consequence of **external** inputs.

Transition and Duration Constraints
For symbolic entities



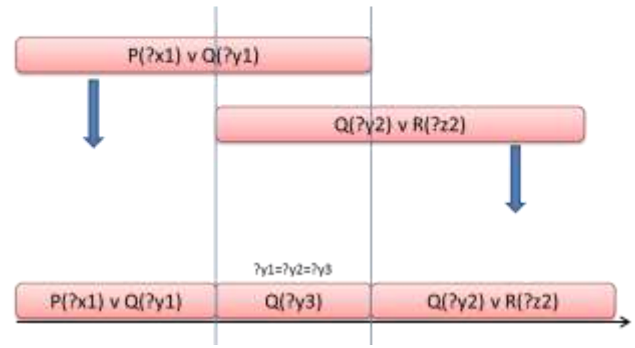


Numerical Bounds or
other Constraints

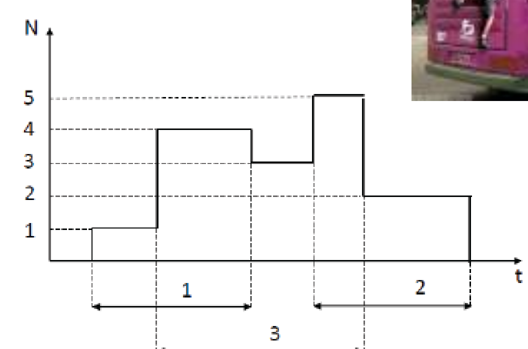


Profile Update

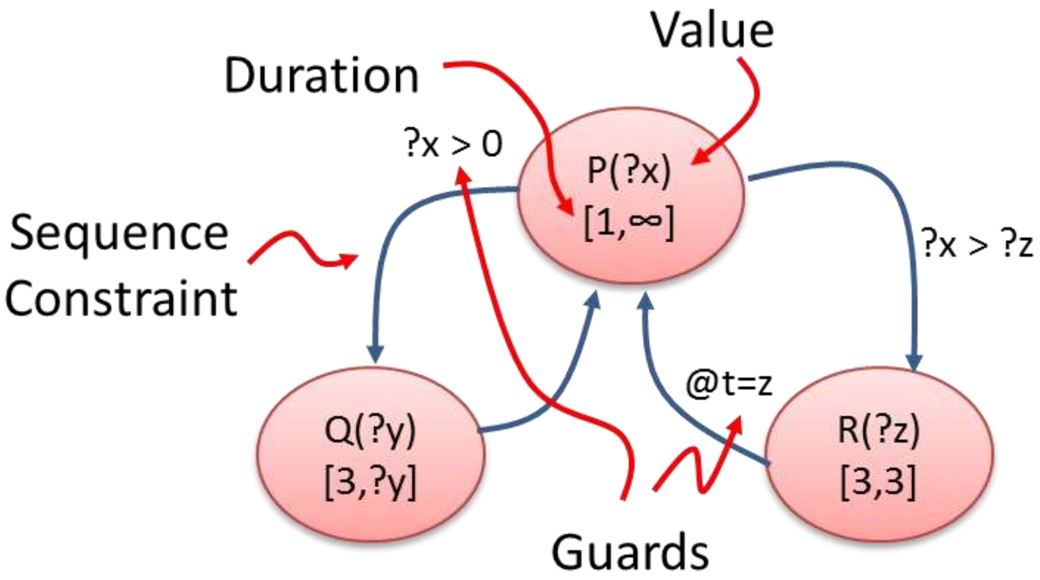
1. The world is modeled as a set of **entities** (physical subsystems)…
2. whose properties can vary in **time**…
3. according with an **internal** logic…
4. or as a consequence of **external** inputs.

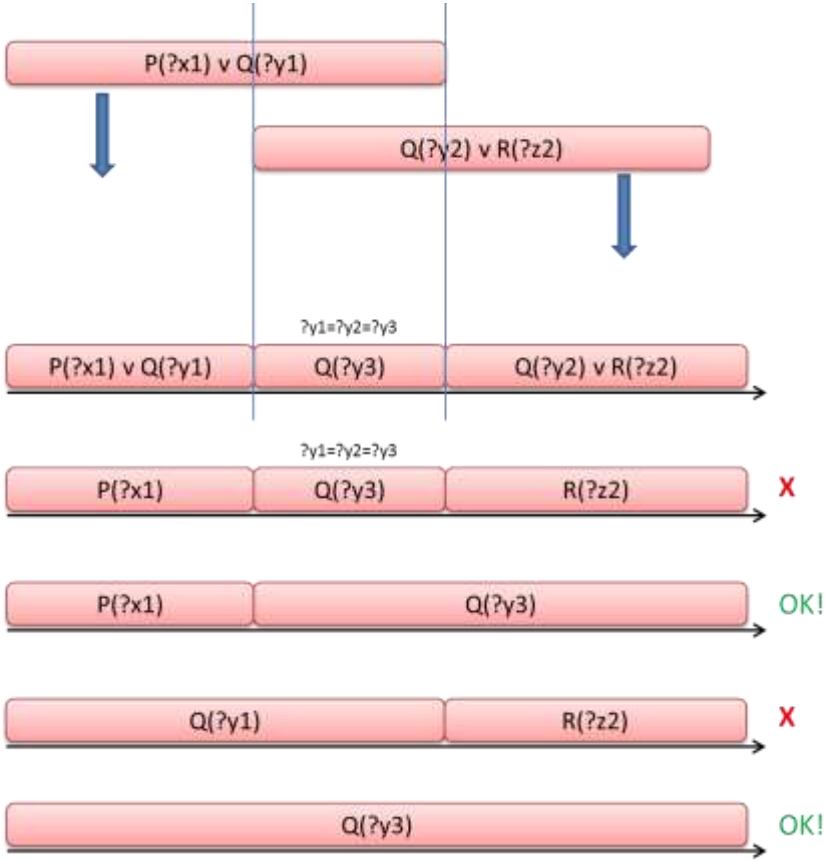


5. Multiple Definitions:
   a. Tokens
   b. Decisions
   c. Interval Constraints
   d. Activities (for numeric components)
   e. Temporal Assertions



State Variables





Resources

1. A resource models a domain feature with a “limited” capacity or availability.

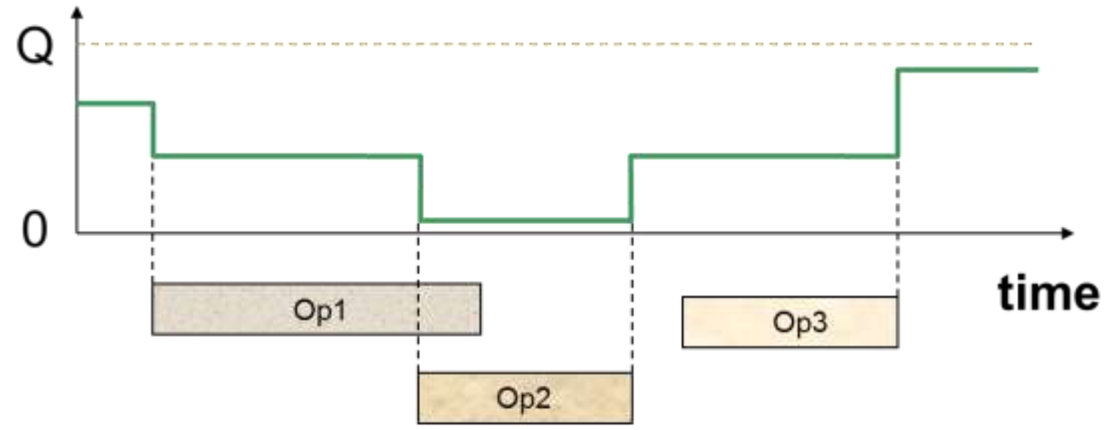
2. Modeled as components whose behaviors are numerical functions of time. Each behavior represents a different profile of resource allocation in time

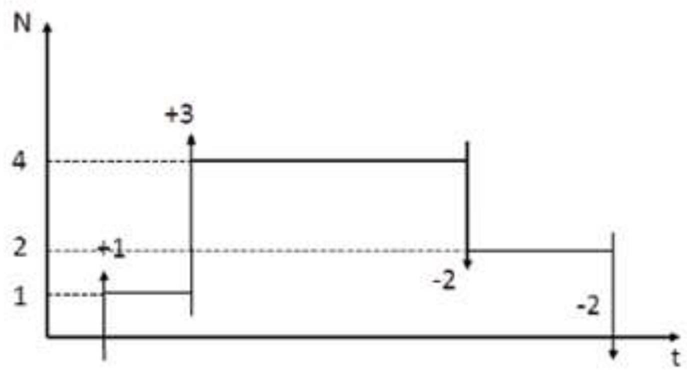
3. The concept of limited availability is defined by means of minimum and maximum value for the behaviors

4. An event capture the concept of resource usage
    a. for a reusable resource an event represents an amount of resource booked on a temporal interval
    b. for a consumable resource an event is an amount produced or consumed in a time instant

Resources

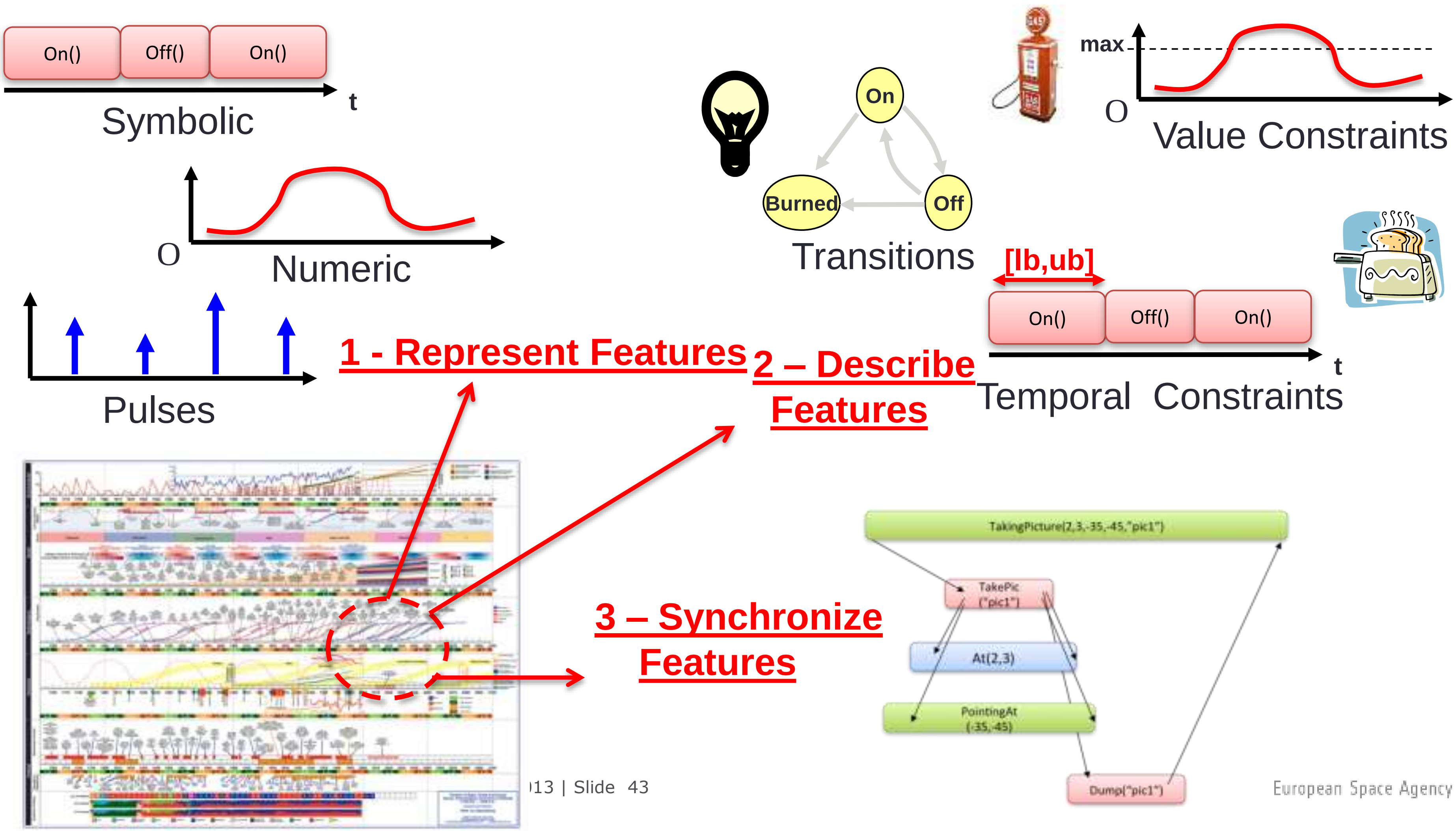
Reusable Resource





Consumable Resource

Timeline-Based Planning



Domain Theory

1. Components in a domain cannot be considered as reciprocally decoupled, rather you need to model the fact that they influence each other's behavior:
   a. defining tuples of valid component behaviors
   b. defining how events have to be synchronized

2. A **domain theory** defines valid patterns of applicable events and/or valid combination of behaviors

3. A **synchronization** is a statement $\rho^r \rightarrow \langle \mathcal{S}, \mathcal{R}_\mathcal{T}, \mathcal{R}_\nu \rangle$:
   a. ρ^r is a temporal property called "reference"
   b. $\mathcal{S} = \{\rho_i^\mathcal{S}\}$ is a set of temporal properties called "supports"
   c. $\mathcal{R}_\mathcal{T}$ is a set of n-ary relations on time frames
   d. $\mathcal{R}_\nu$ is a set of n-ary relations on values

Synchronizations

1. A synchronization can be **applicable** to a timeline or to a set of events

2. A synchronization applicable to a timeline or to a set of events can be **satisfied** by the timelines and events

3. A **Domain theory** DT is a set of synchronizations

4. A domain theory DT is **satisfied** by a set of timelines and events if any applicable synchronization in is satisfied

$$\mathcal{S}at(\mathcal{DT}, \mathcal{TL}, \mathcal{E}v) = \top \leftrightarrow$$

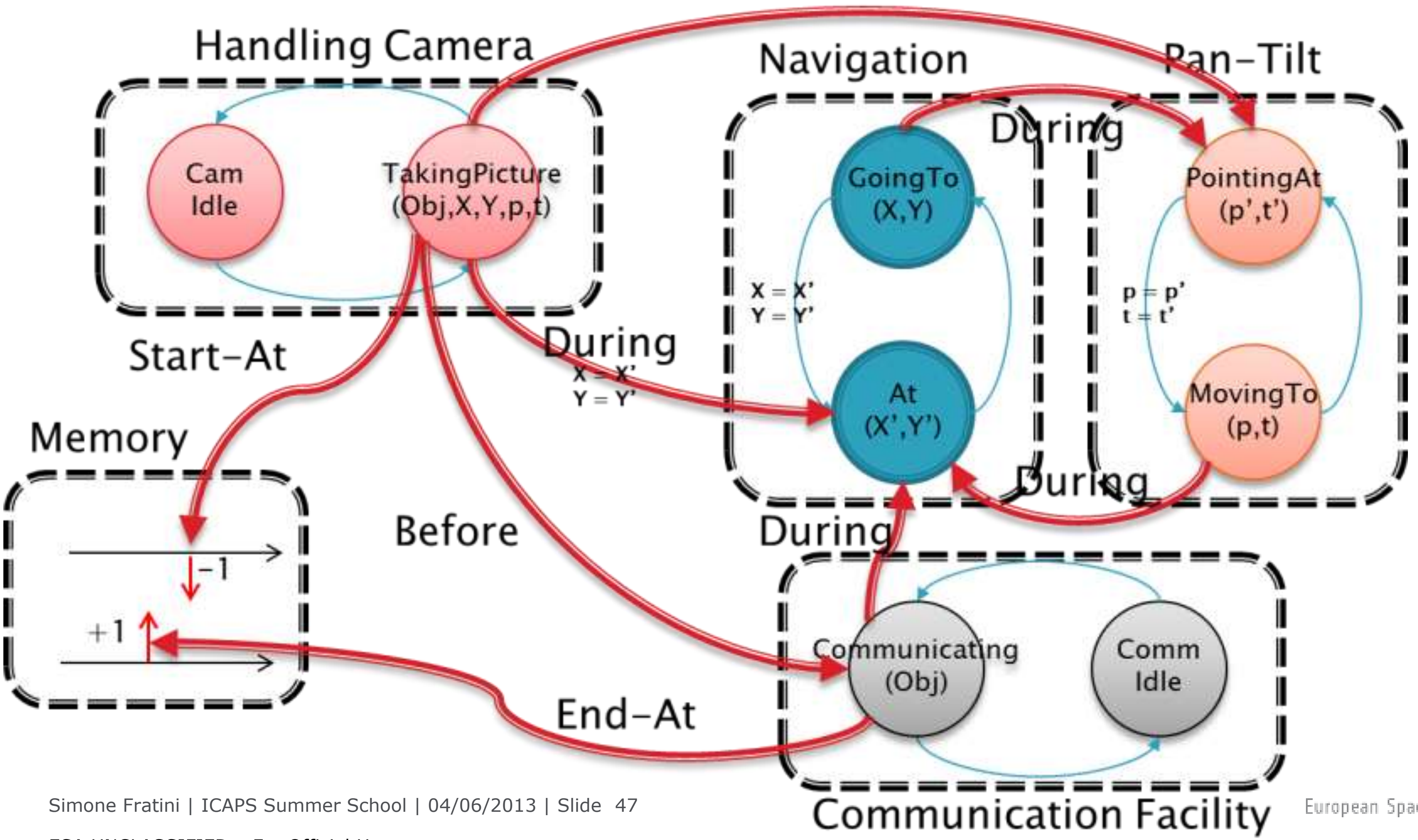
$$\forall \gamma \in \Gamma, \mathcal{A}pp(\gamma, \mathcal{TL}, \mathcal{E}v) \rightarrow \mathcal{S}at(\gamma, \mathcal{TL}, \mathcal{E}v)$$

Example: A Robotic Domain…

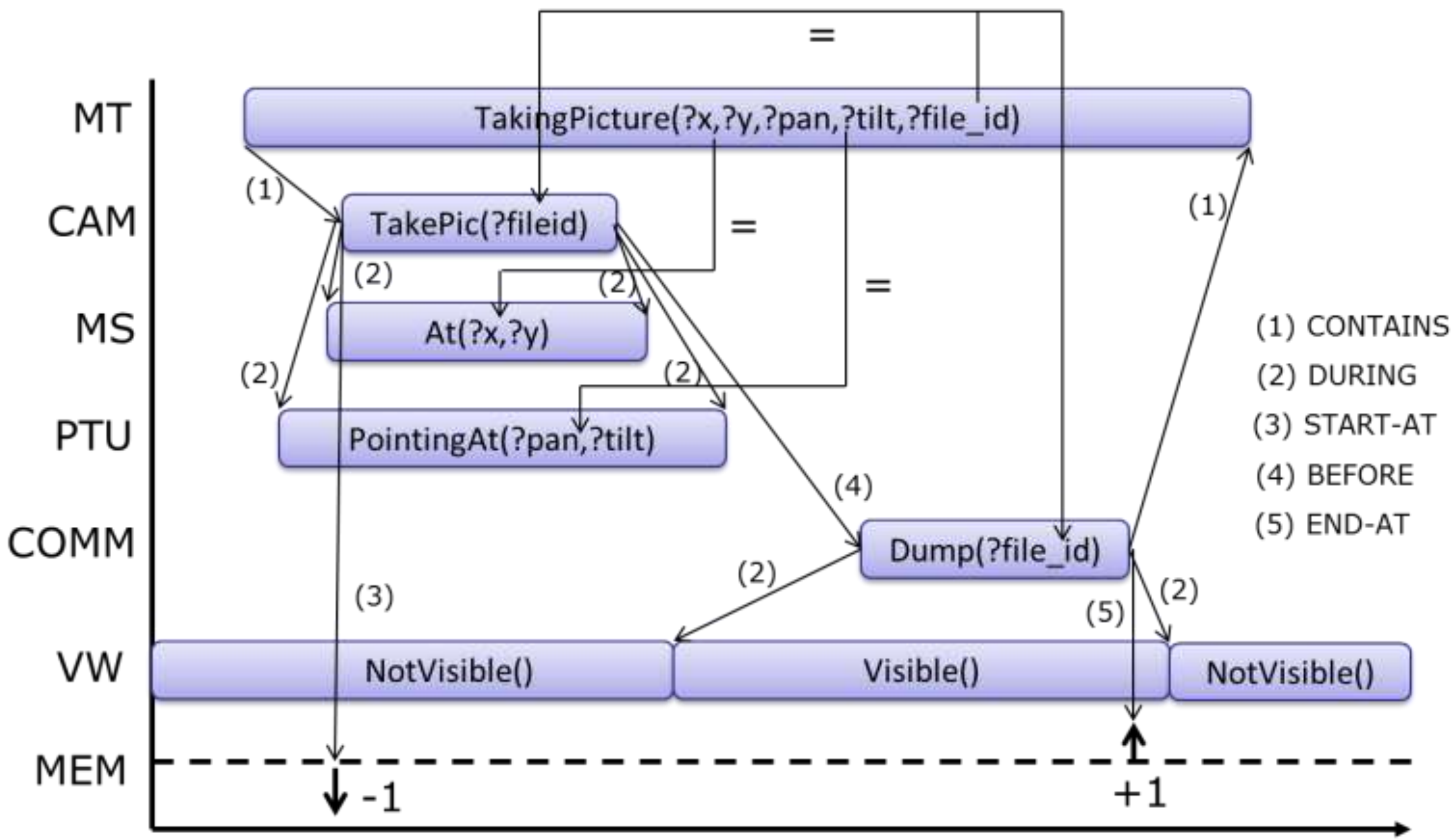
1. A robotic Platform equipped with:
    a. A mobility system
    b. A pan-tilt unit (PTU)
    c. Two stereo cameras (mounted on top of PTU)
    d. A communication facility
    e. An onboard memory

2. Robot has to:
    a. Autonomously navigate the environment
    b. Move the PTU
    c. Take pictures
    d. Store and dump taken pictures when possible

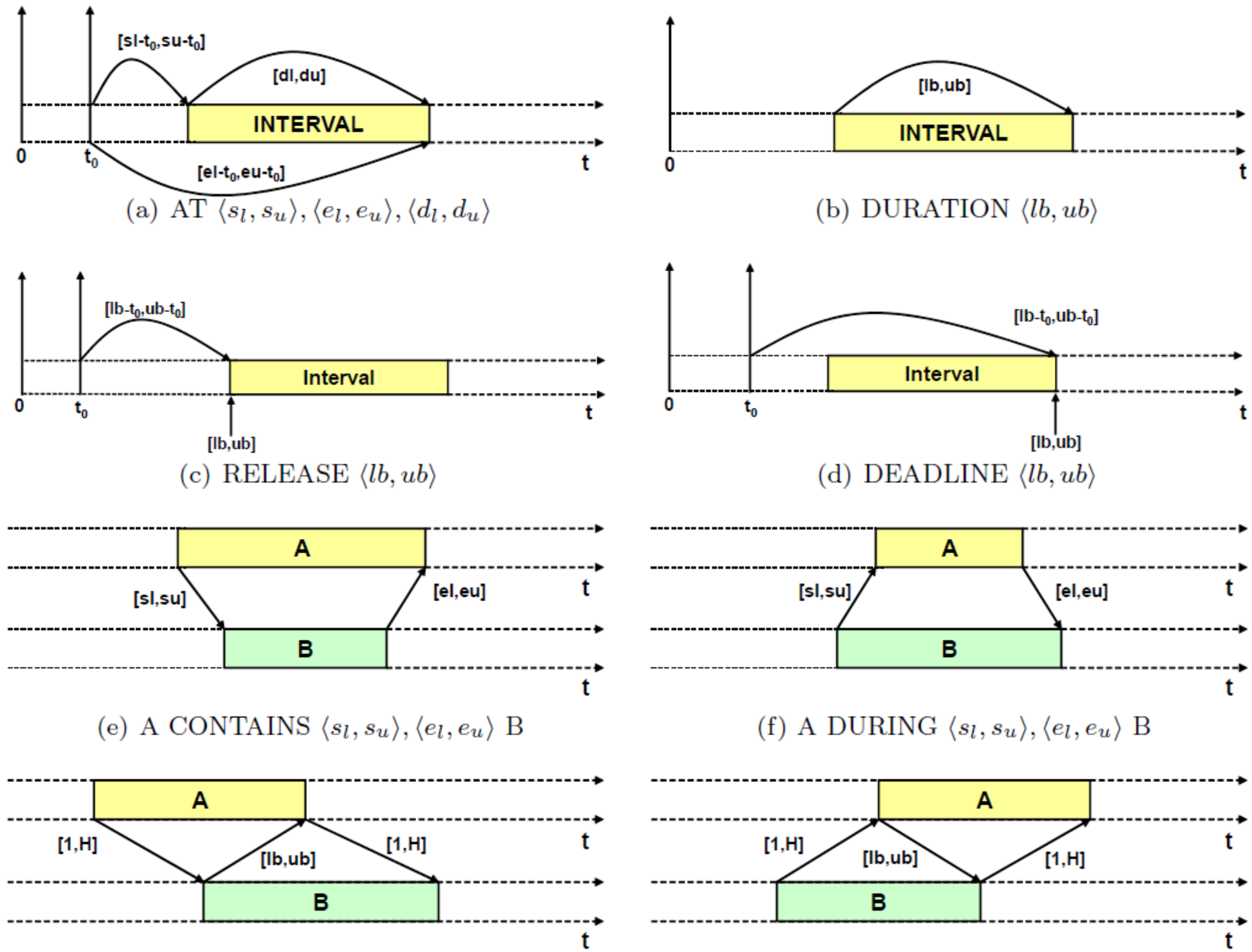
Example – Robotic Domain



Example - Synchronization



Example of Temporal Relations



(a) AT $\langle s_l, s_u\rangle, \langle e_l, e_u\rangle, \langle d_l, d_u\rangle$

(b) DURATION $\langle lb, ub\rangle$

(c) RELEASE $\langle lb, ub\rangle$

(d) DEADLINE $\langle lb, ub\rangle$

(e) A CONTAINS $\langle s_l, s_u\rangle, \langle e_l, e_u\rangle$ B

(f) A DURING $\langle s_l, s_u\rangle, \langle e_l, e_u\rangle$ B

(g) A OVERLAPS $\langle lb, ub\rangle$ B

(h) A OVERLAPPED-BY $\langle lb, ub\rangle$ B

Example – DDL.3 Specification

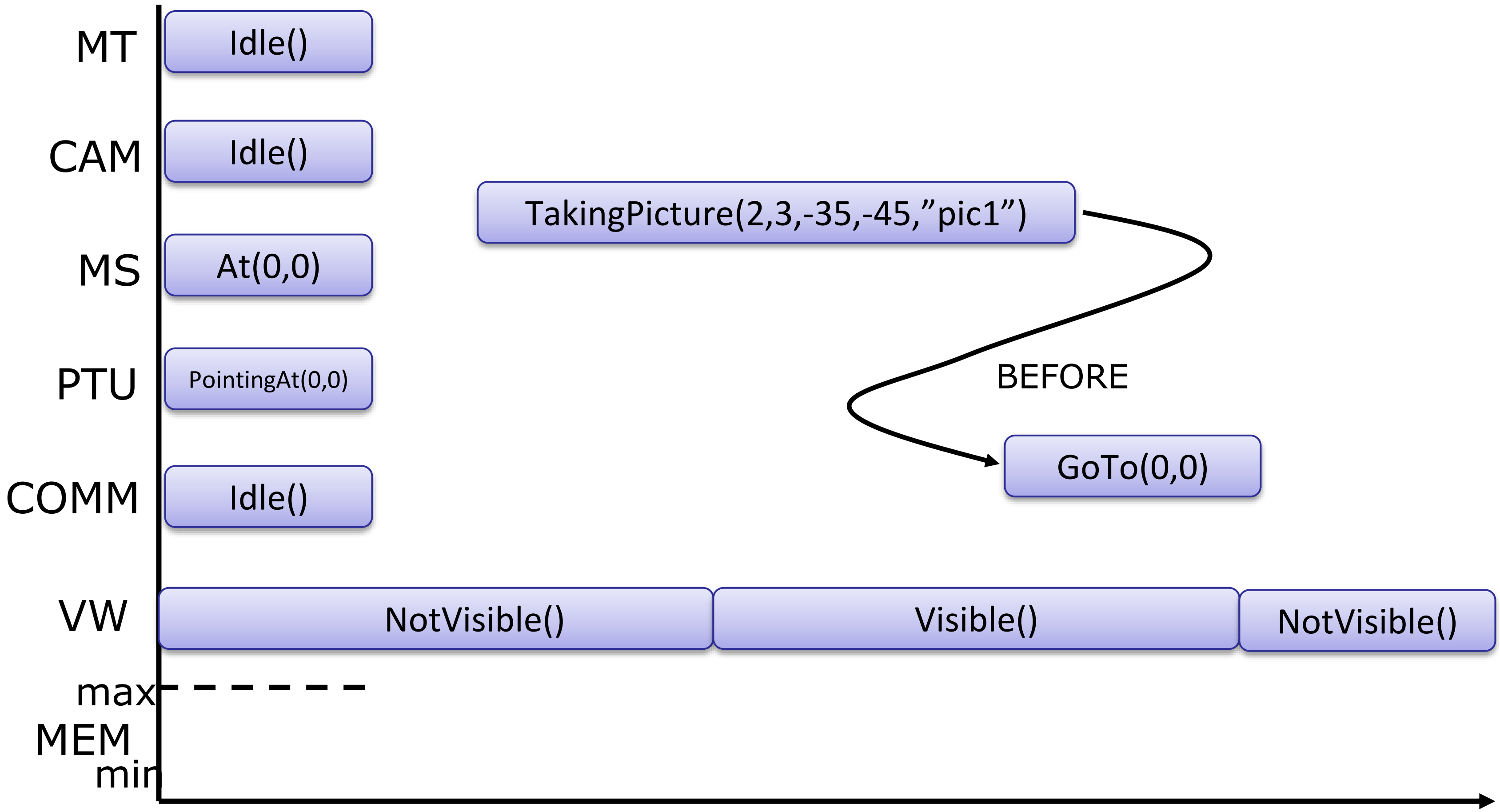
```
SYNCHRONIZE MissionTimeline.mission_timeline {
    VALUE TakePicture(?file_id1, ?x1, ?y1, ?pan1, ?tilt1) {
      cd1 Camera.camera.TakingPicture(?file_id2, ?x2, ?y2, ?pan2, ?tilt2);
      cd2 Communication.communication.Communicating(?file_id3);
      CONTAINS [0,+INF] [0,+INF] cd1;
      CONTAINS [0,+INF] [2,2] cd2;
      cd1 BEFORE [0, +INF] cd2;
      ?x1 = ?x2; ?y1 = ?y2;
      ?pan1 = ?pan2; ?tilt1 = ?tilt2;
      ?file_id1 = ?file_id2; ?file_id1 = ?file_id3; }}

SYNCHRONIZE Camera.camera {
    VALUE TakingPicture(?file_id1, ?x, ?y, ?pan, ?tilt) {
      cd3 Platine.platine.PointingAt(?pan, ?tilt);
      cd2 RobotBase.robot_base.At(?x, ?y);
      cd1 Mem.profile.Consume(1);
      DURING [0, +INF] [0, +INF] cd2;
      DURING [0, +INF] [0, +INF] cd3;
      START-AT cd1;}}

SYNCHRONIZE Communication.communication {
    VALUE Communicating(?file_id1) {
      cd4 RobotBase.robot_base.At(?x2, ?y2);
      cd1 CommunicationVW.communication_windows.Visible();
      cd2 Mem.profile.Produce(1);
      DURING [0, +INF] [0, +INF] cd4;
      DURING [0, +INF] [0, +INF] cd1;
      END-AT cd2;}}
```

Problem Definition

A problem can be defined in general by means of:

1. a set of timelines $\mathcal{TL}_0$ stating the initial system status
2. a pair $\langle \mathcal{TL}_g, \mathcal{E}V_g \rangle$ stating the goal conditions
3. a domain theory $\mathcal{DT}$ specifying the properties required for valid timelines and events applicable to the system

Example – Problem Definition



Example – Problem Definition

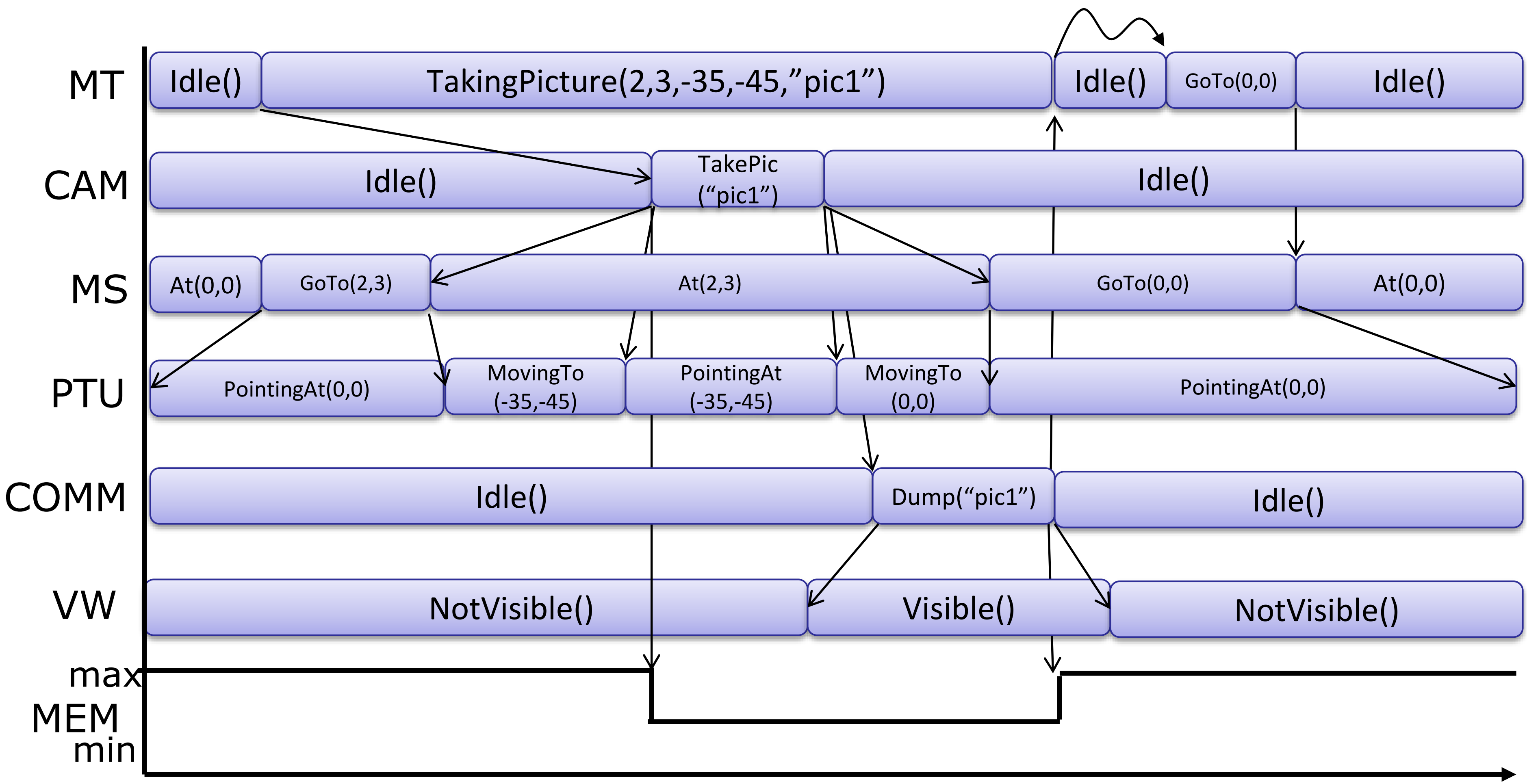
```
PROBLEM GOAC_Single_Reactor_Init (DOMAIN GOAC_Single_Reactor) {

mtl0 <fact> MissionTimeline.mission_timeline.Idle() AT [0, 0] [1, +INF] [1, +INF];

comvw1 <fact> CommunicationVW.communication_windows.Visible()
                 AT [10, 10] [1000, 1000] [990, 990];
comvw2 <fact> CommunicationVW.communication_windows.Visible()
                 AT [1200, 1200] [3000, 3000] [1800, 1800];

pos0 <fact> RobotBase.robot_base.At(?x1 = 0, ?y1 = 0) AT [0, 0] [1, +INF] [1, +INF];
or0 <fact> Platine.platine.PointingAt(?pan1 = 0, ?tilt1 = 0) AT [0, 0] [1, +INF] [1, +INF];
cam0 <fact> Camera.camera.CamIdle() AT [0, 0] [1, +INF] [1, +INF];
com0 <fact> Communication.communication.CommIdle() AT [0, 0] [1, +INF] [1, +INF];

tp1 <goal> MissionTimeline.mission_timeline.TakePicture(1,2,3,-35,-45);
tp2 <goal> MissionTimeline.mission_timeline.GoTo(0,0);
tp1 BEFORE [1, +INF] tp2;
}
```

Solution Definition

The goal of the solver is to identify an set of events $\mathcal{E}V_s$ subset of $\mathcal{E}V_g$ such that:

1. allows an extraction of a timeline from $\mathcal{TL}_0$
2. the timeline extracted $\mathcal{TL}_s$ describes a subset of $\mathcal{TL}_g$
3. $\langle \mathcal{TL}_s, \mathcal{E}V_s \rangle$ satisfies the domain theory

Example - Solution



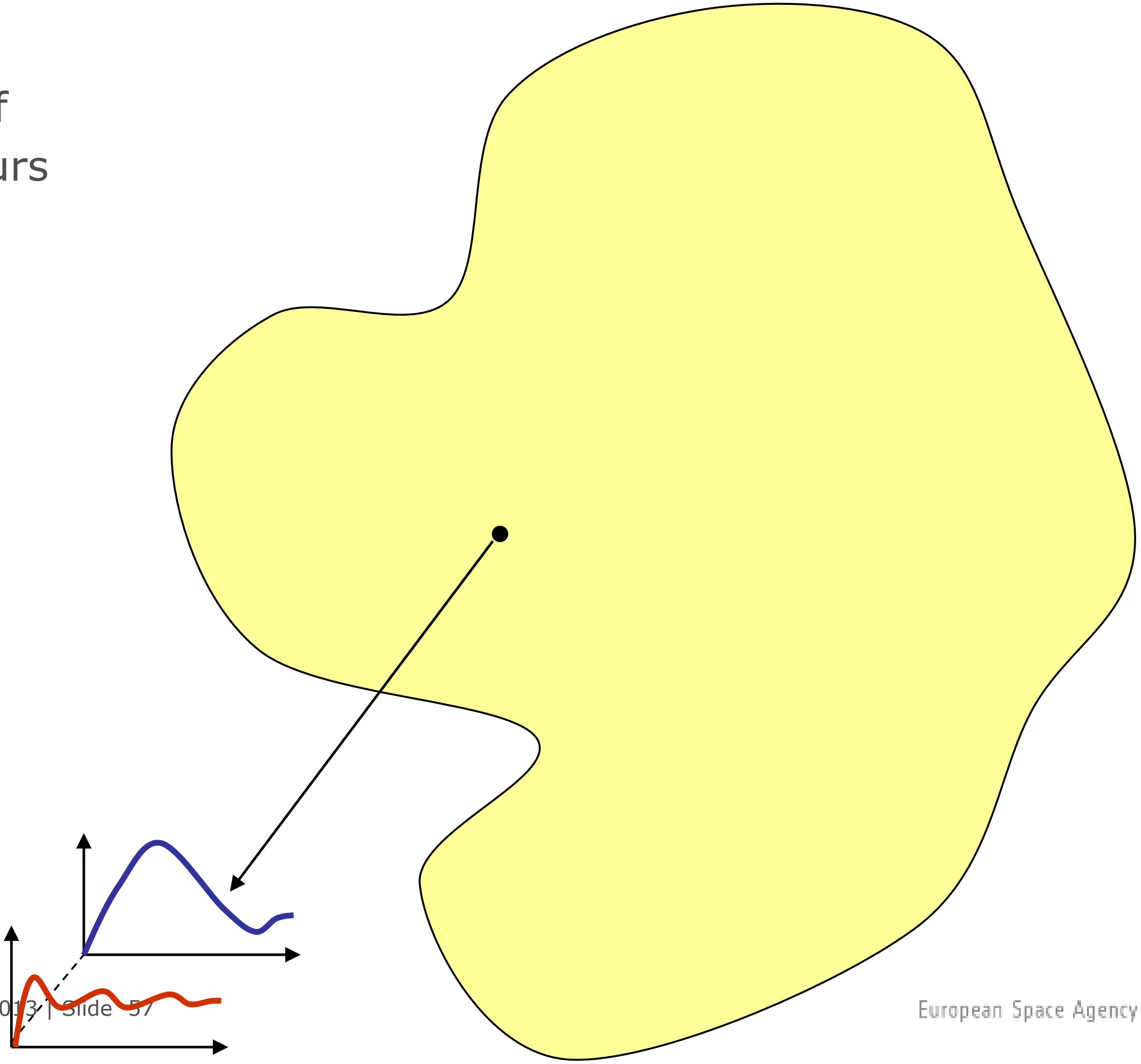
Search Space Representation:

Envelopes

Types of timelines

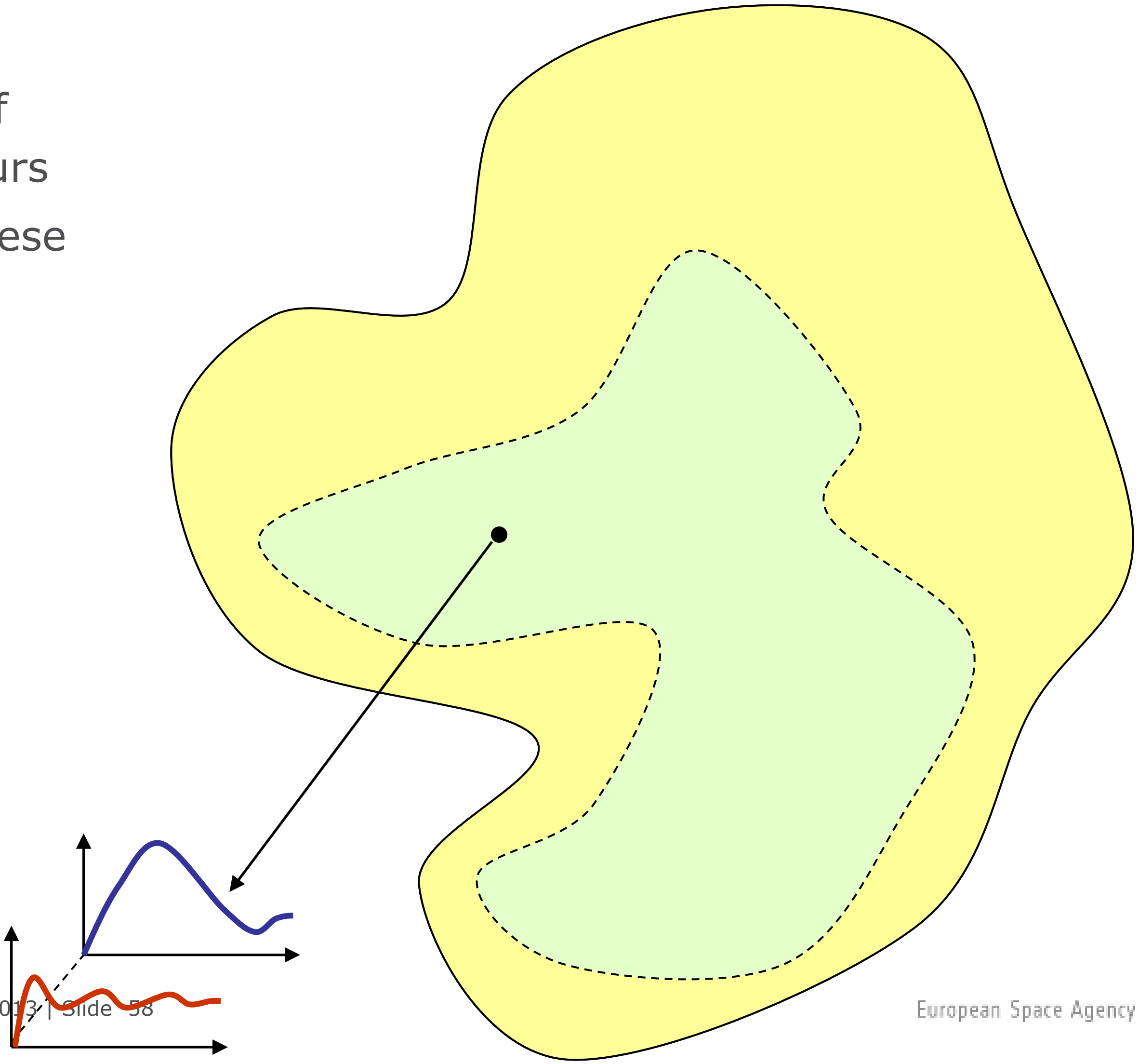
The Envelope

1. Search in the space of components’ behaviours



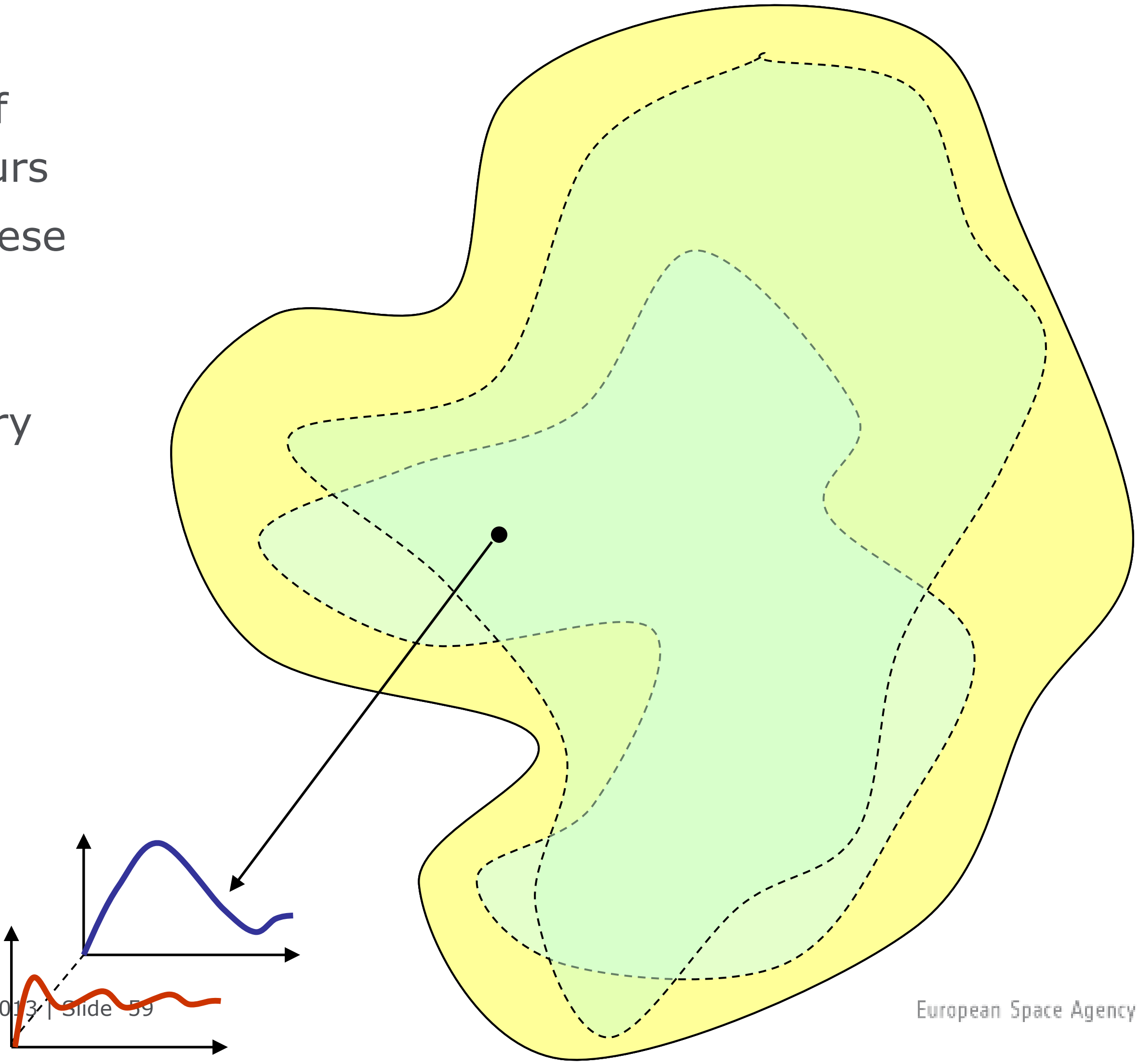
The Envelope

1. Search in the space of components' behaviours
2. Define a sub-set of these behaviours as valid behaviours



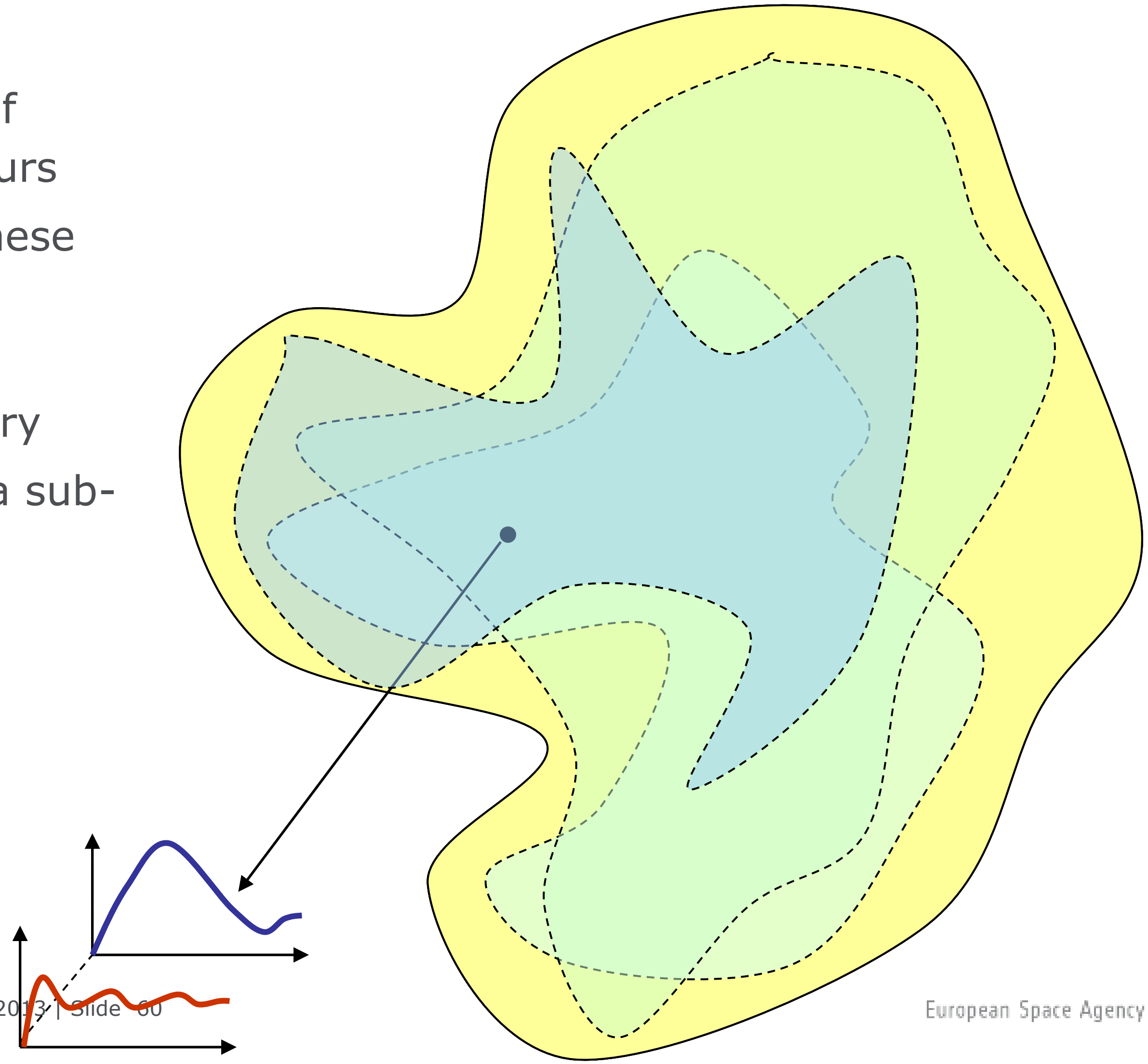
The Envelope

1. Search in the space of components' behaviours
2. Define a sub-set of these behaviours as valid behaviours
3. Define a domain theory



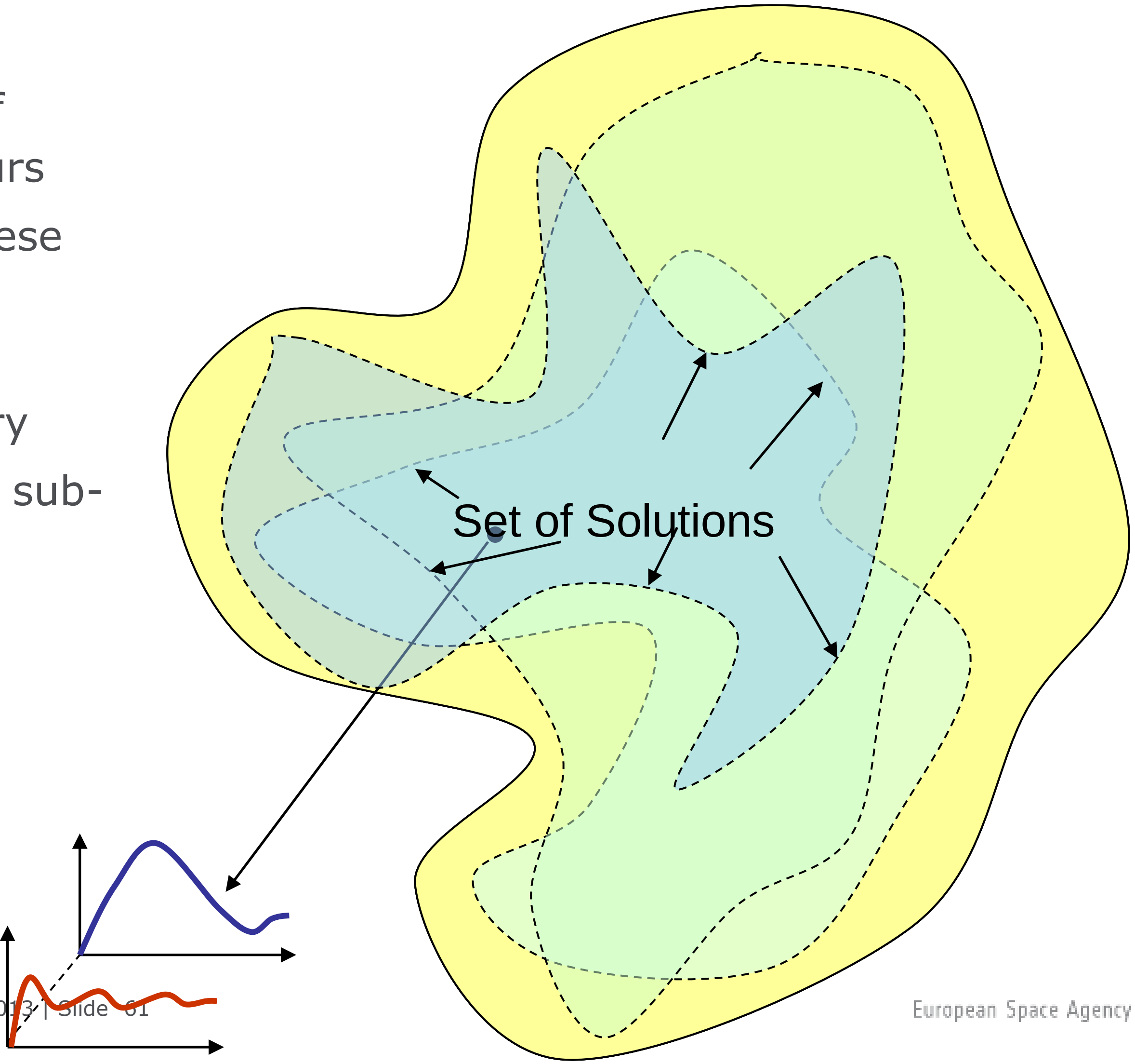
The Envelope

1. Search in the space of components' behaviours
2. Define a sub-set of these behaviours as valid behaviours
3. Define a domain theory
4. Define a problem as a sub-set of behaviors



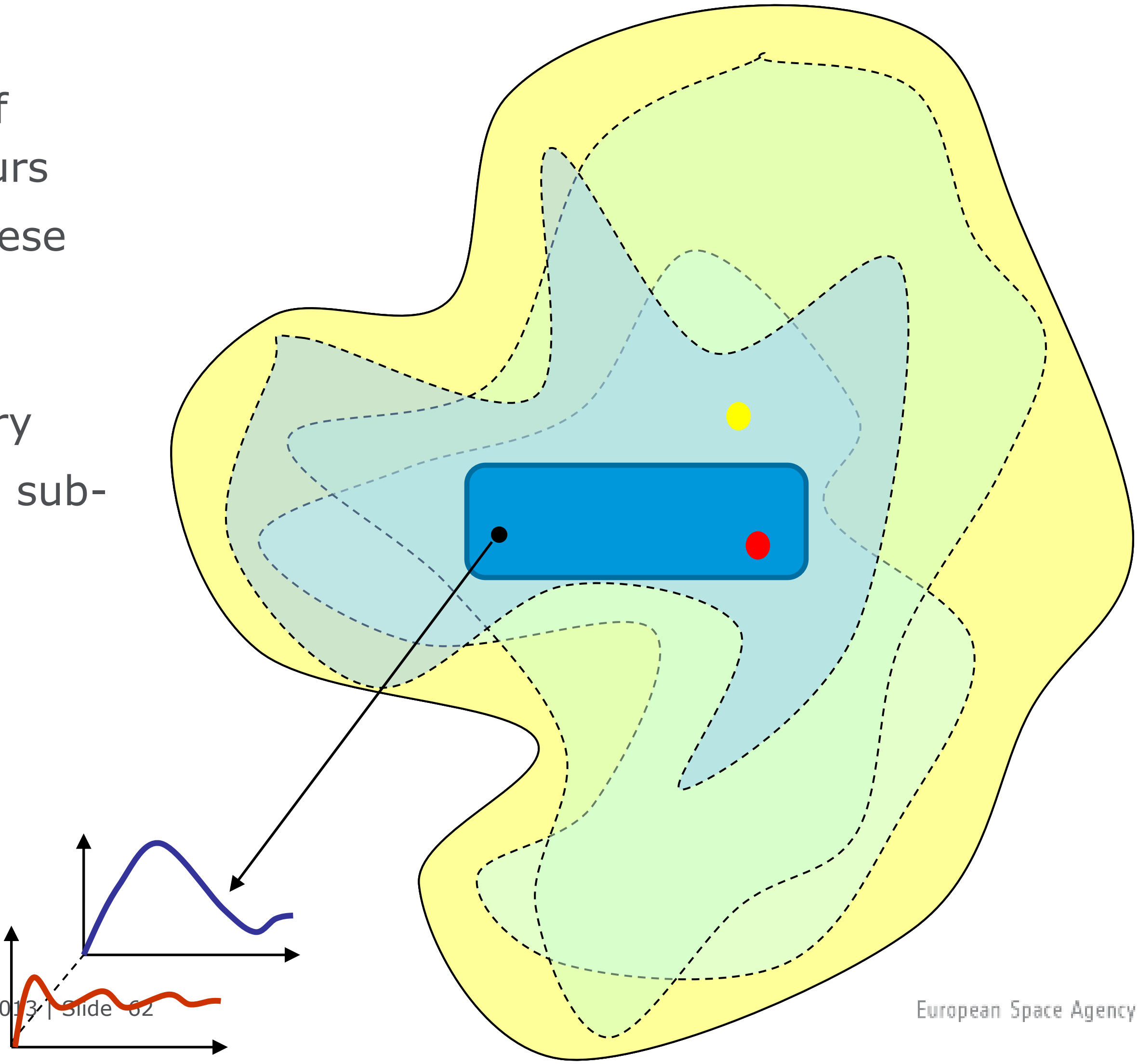
The Envelope

1. Search in the space of components’ behaviours
2. Define a sub-set of these behaviours as valid behaviours
3. Define a domain theory
4. Define a problem as a sub-set of behaviours



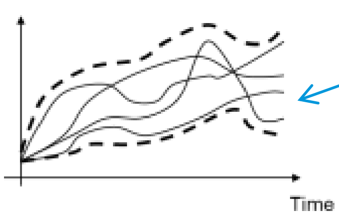
The Envelope

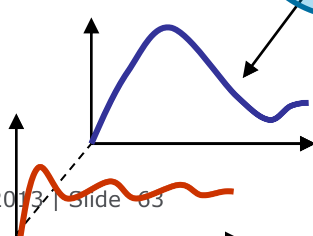
1. Search in the space of components' behaviours
2. Define a sub-set of these behaviours as valid behaviours
3. Define a domain theory
4. Define a problem as a sub-set of behaviours
5. Identify the Kernel…

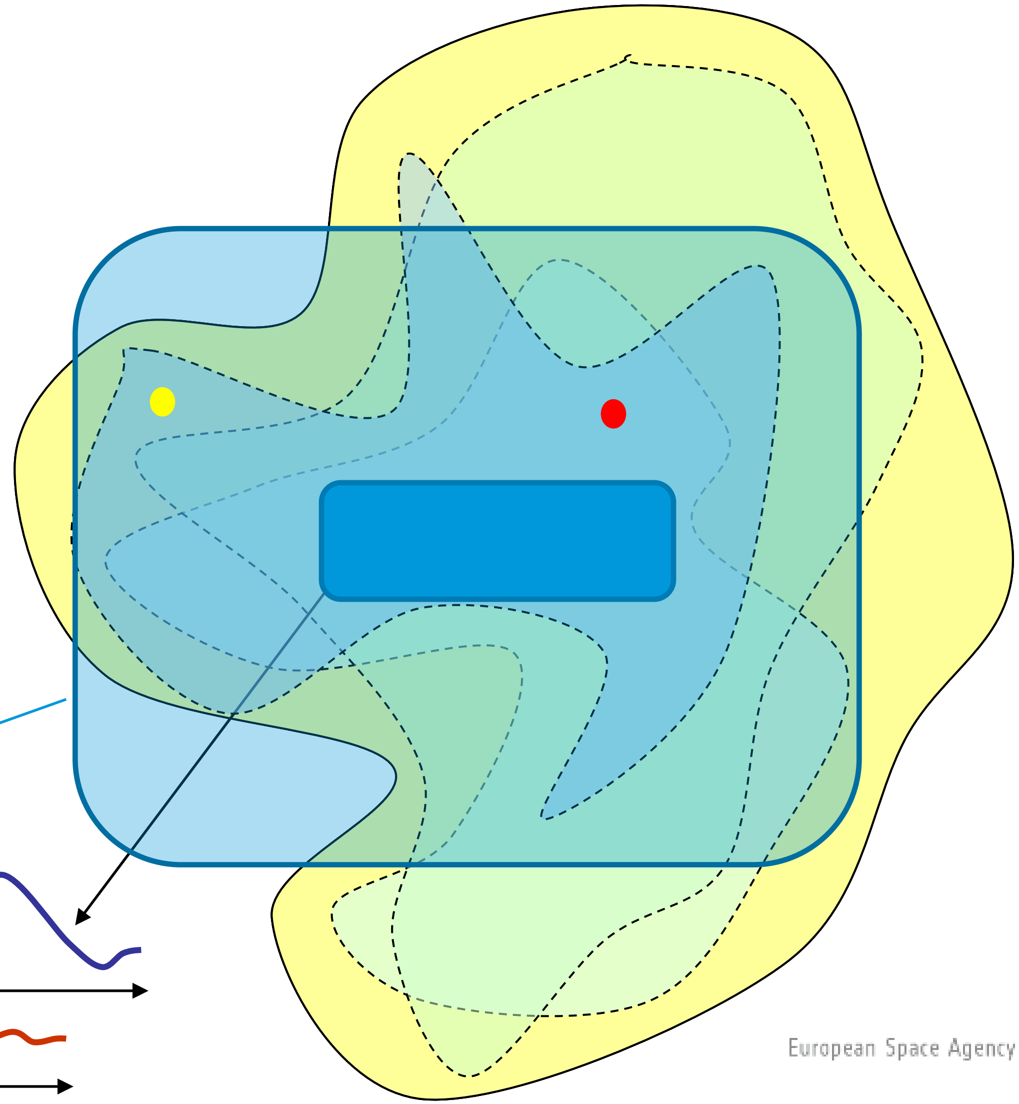


The Envelope

1. Search in the space of components' behaviours
2. Define a sub-set of these behaviours as valid behaviours
3. Define a domain theory
4. Define a problem as a sub-set of behaviours
5. Identify the Kernel… or the Envelope



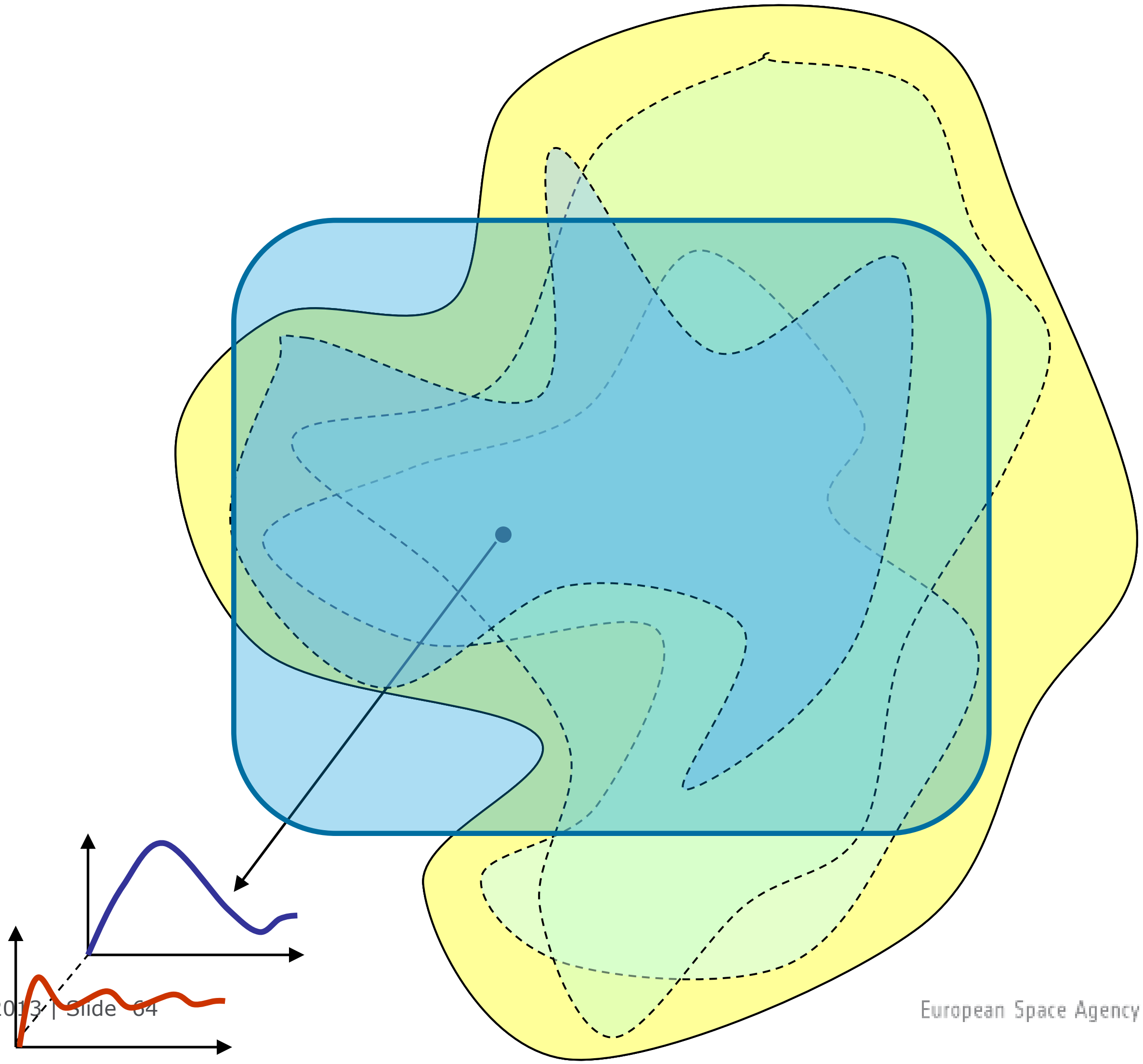




The Envelope

The Envelope:

1. Easy to describe

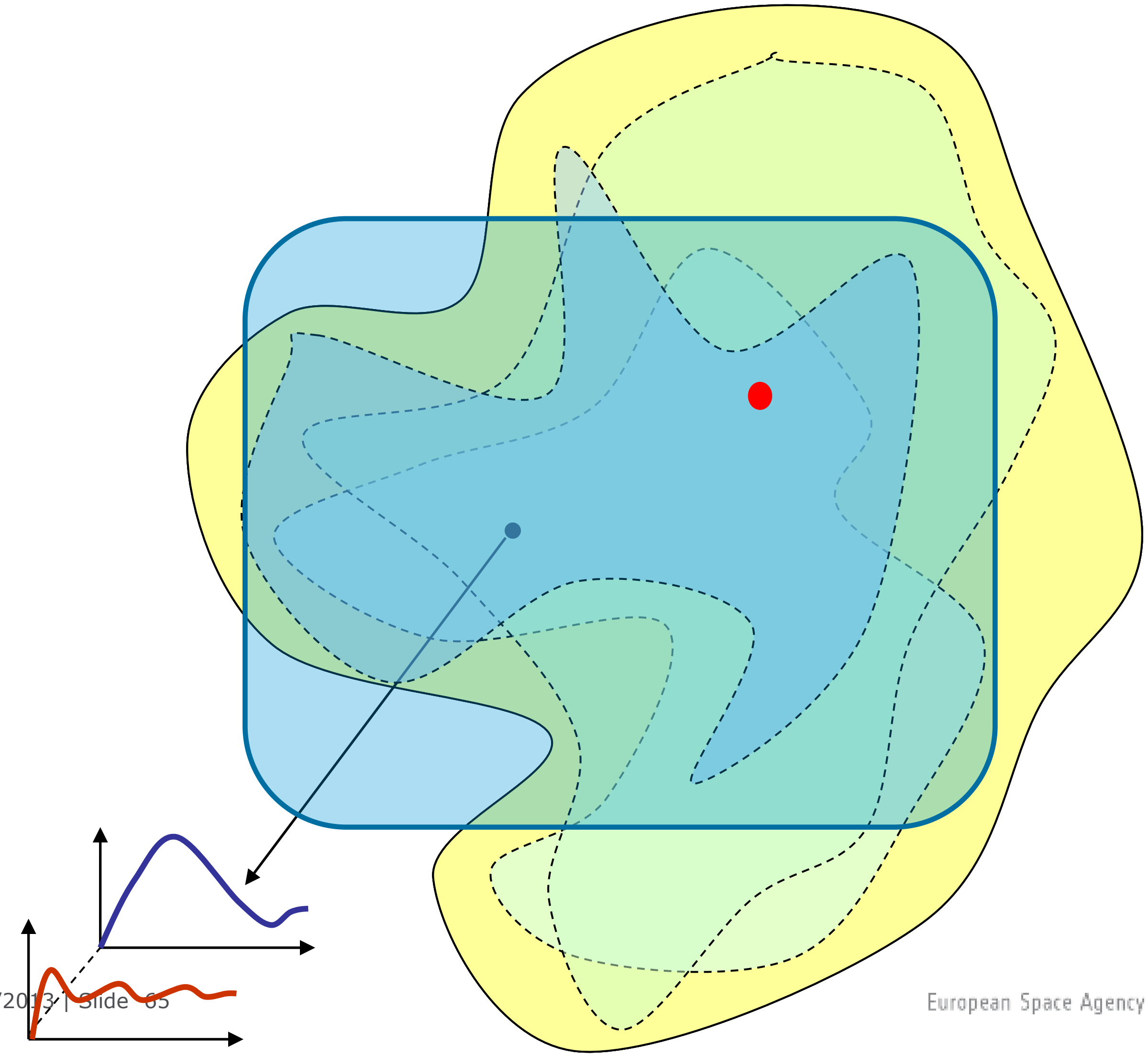


The Envelope

The Envelope:

1. Easy to describe

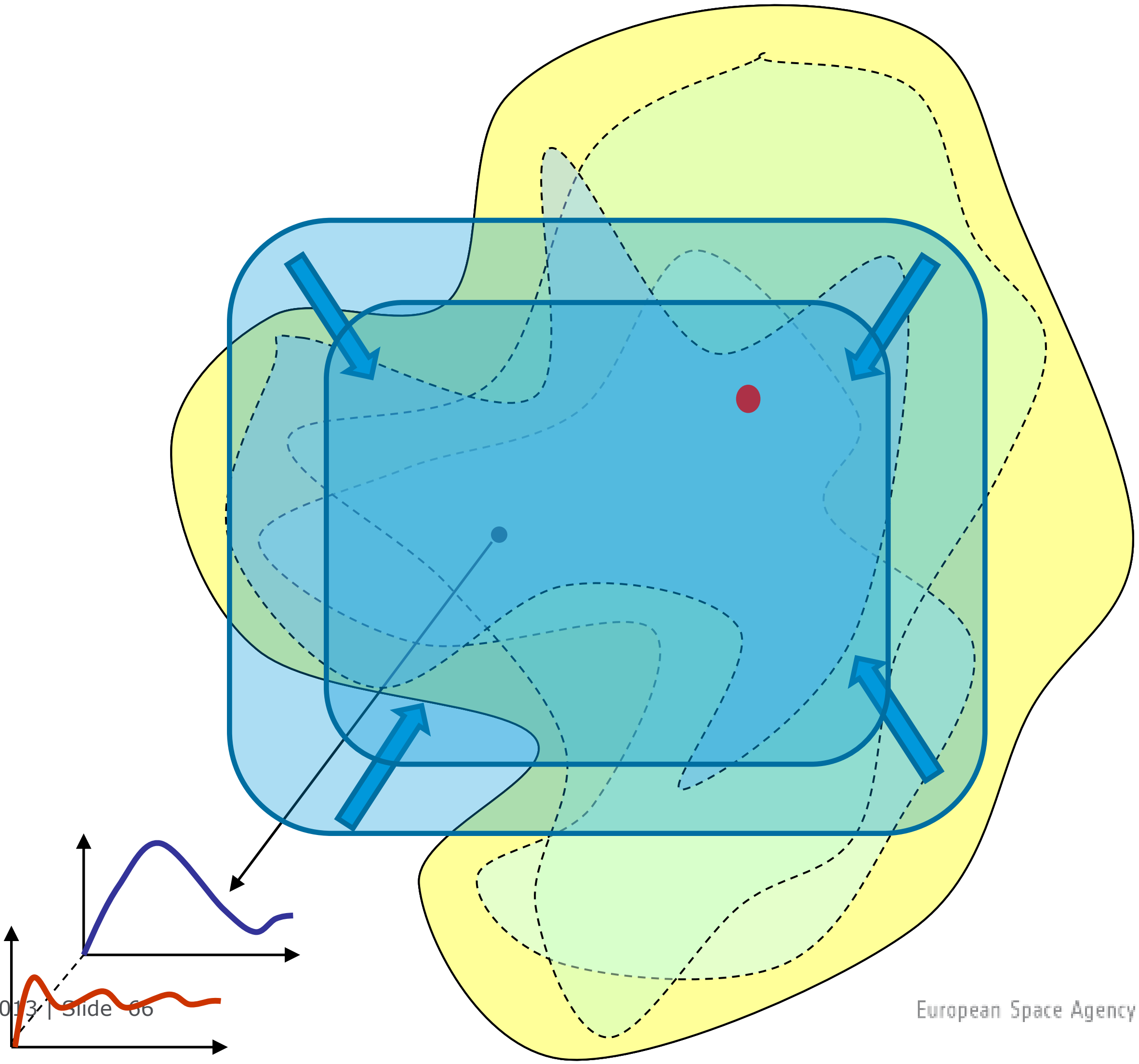
2. Easy to identify valid solutions



The Envelope

The Envelope:

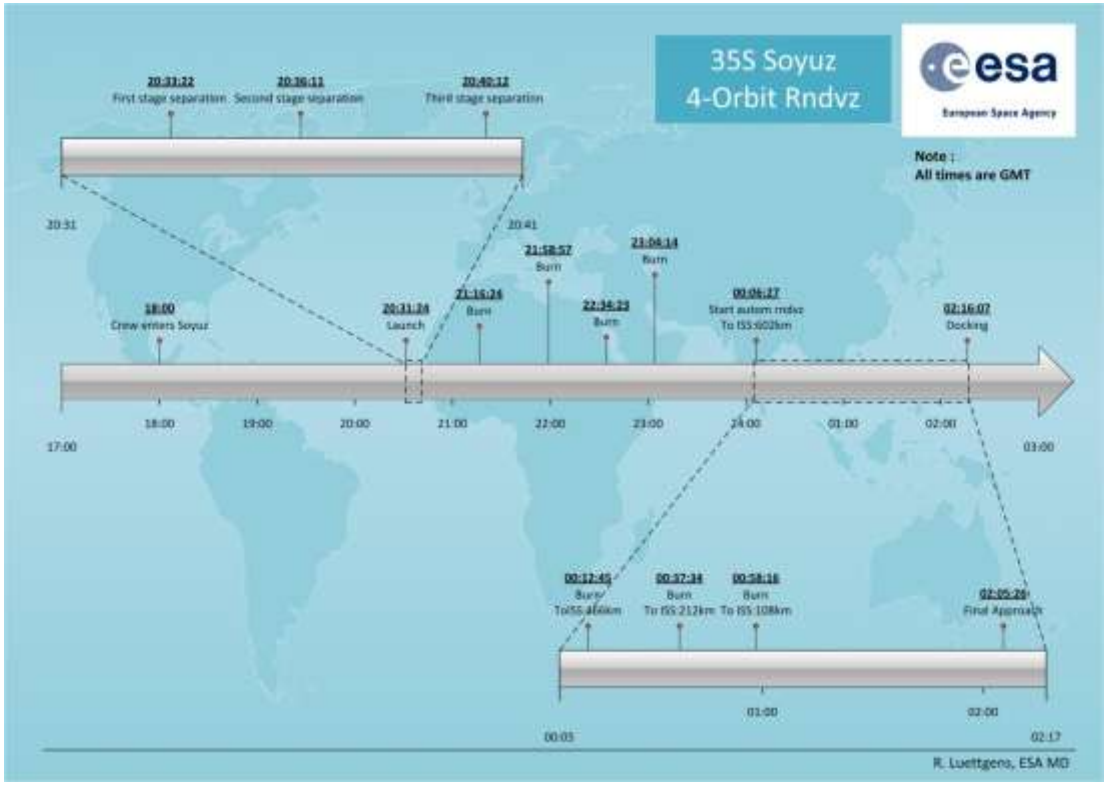
1. Easy to describe
2. Easy to identify valid solutions
3. Easy to narrow during search



Timelines

Timeline (n): a graphical representation of a period of time, on which important events are marked. - Oxford English Dictionary

A timeline is a way of displaying a list of events in chronological order, sometimes described as a project artifact. It is typically a graphic design showing a long bar labeled with dates alongside itself and (usually) events labeled on points where they would have happened. - Wikipedia



Timeline?

Less simple that it may look like:

[Muscettola, N. (1994a). HSTS: Integrating Planning and Scheduling. In Zweben, M. and Fox, M.S., editor, *Intelligent Scheduling*. Morgan Kauffmann]

[Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote agent: to go boldly where no AI system has gone before. Artificial Intelligence, 103(1–2):5–48.]

[Frank, J., and Joonsson, A. 2003. Constraint-based attribute and interval planning. *Constraints* 8(4):339–364.]

[A. Cesta, S. Fratini and A. Oddi. Planning with Concurrency, Time and Resources: A CSP-Based Approach. In I. Vlahavas, D. Vrakas (Eds.), "Intelligent Techniques for Planning", Idea Group Publishing, pp. 259-295, 2005]

[Cédric Pralet, Gérard Verfaillie: Using Constraint Networks on Timelines to Model and Solve Planning and Scheduling Problems. ICAPS 2008: 272-279]

[Fratini, S.; Pecora, F.; and Cesta, A. 2008. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. *Archives of Control Sciences* 18(2):231–271]

[Verfaillie, G.; Pralet, C.; and Lematre, M. 2010. How to model planning and scheduling problems using constraint networks on timelines. *Knowledge Eng. Review* 25(3):319–336]

[Chien, S.; Johnston, M.; Frank, J.; Giuliano, M.; Kavelaars, A.; Lenzen, C.; and Policella, N. 2012. A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations. In Proceedings of Conference on Space Operations]

[Fratini, S., and Cesta, A. 2012. The APSI Framework: A Platform for Timeline Synthesis. In Proceedings of the 1st Workshops on Planning and Scheduling with Timelines at ICAPS-12, Atibaia, Brazil.]

[Frank, J. What is a Timeline? In Proceedings of the 4th Workshops on Knowledge Engineering for Planning and Scheduling at ICAPS-13, Rome, Italy]

Definitions

- Modelling Primitive: A timeline provides a mean to describe the evolution of a system as a set of parallel histories (or concurrent threads) over linear and continuous time

- Service Provider: A timeline provides a set of services to allow a solver to interact with the computed history of the evolution of a system:
    a. Detect violations over time intervals
    b. Insert/remove data in time
    c. Check whether an insertion violates a constraint
    d. Return a list of valid times to place data
    e. Invoke custom code to calculate complex data
    f. …

Definitions (2)

- Functional Definition: A timeline is a data structure to entail powerful reasoning on data over time (like Graphplan for PDDL for instance)

- CSP Definition: A timeline is a constraint network on time and data

- Mathematical definitions:

  A timeline T L for over a temporal interval H is a tuple $<T,V,f^T,f^N>$ where:
  - T is a finite, ordered set of time variables ranging in H=[0,H);
  - V is a set of variables ranging in D such that $|V| = |T|$;
  - $f^T \subseteq H^{|T|}$ is a set of possible temporal occurrence assignment to variables in T such that for each assignment $<t_0,\ldots, t_i,\ldots, t_H> \in T$, $t_0 = 0$, $t_H = H$ and $t_i < t_j$, $\forall i < j$.
  - $f^N \subseteq V^{|T|}$ is a set of possible assignment of values in V to variables in T

Classification

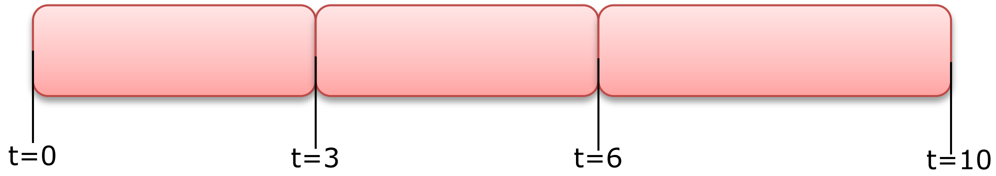
1. Temporal Model:
    a. Fix
    b. Bounded
    c. Flexible, STN with Total Order
    d. Flexible, STN generic
    e. Flexible, DTN

2. Data Model:
    a. Ground
    b. Parameterized
    c. Multi-Valued
    d. Generic

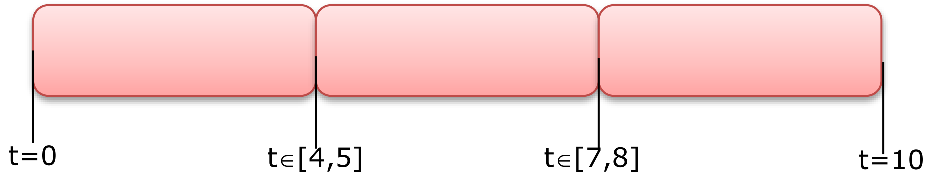
3. Transition Model:
    a. Markovian
    b. Global

Temporal Model Classification

1. Fix

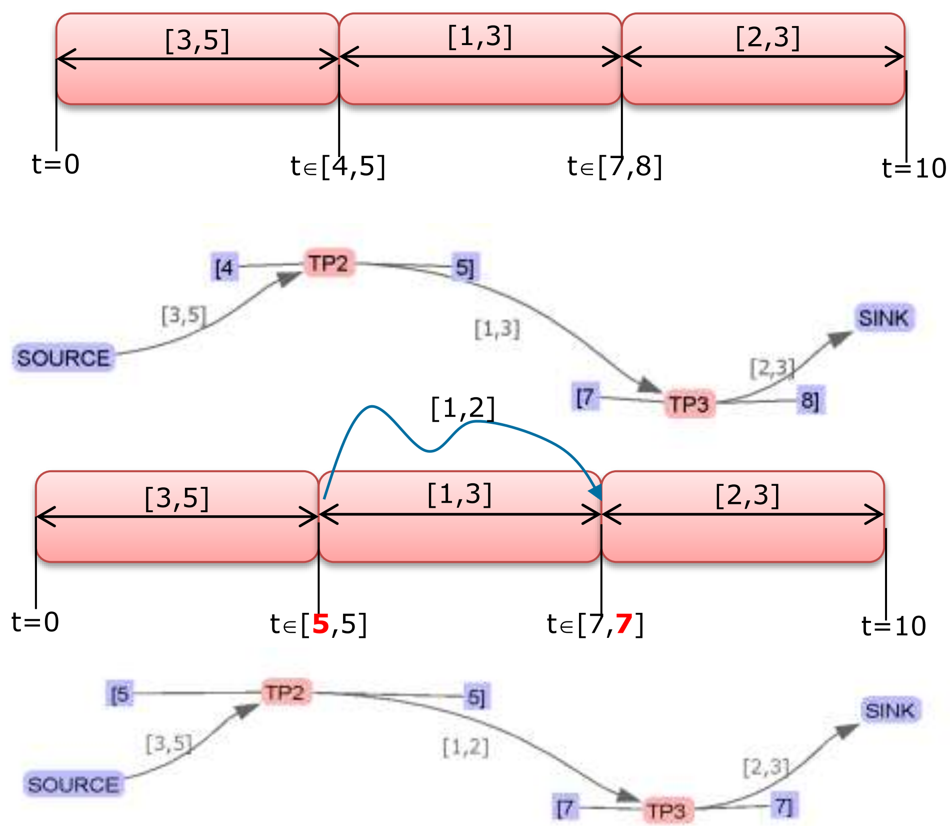


2. Bounded



Temporal Model Classification

3. Flexible, STN with Total Order



The Envelope:

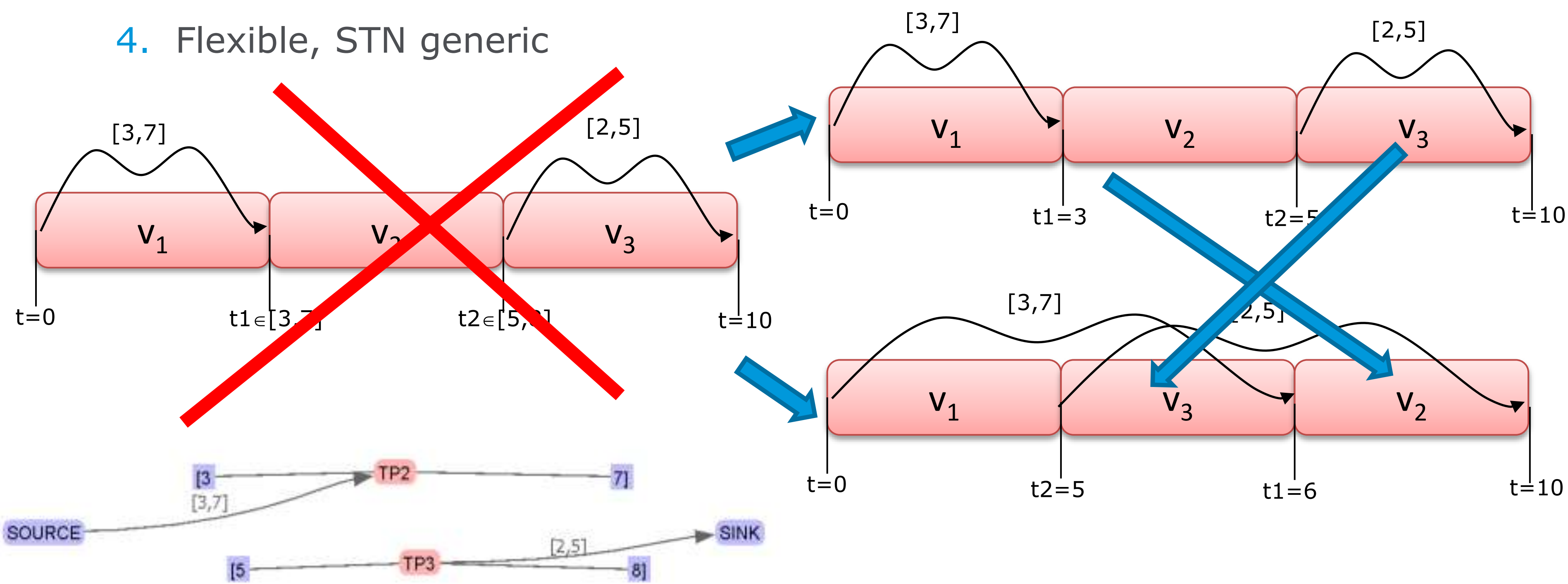
1. Easy to describe
2. Easy to identify valid solutions
3. Easy to narrow during search

1. t_0, t_1, t_2, t_H + $[t_0,t_1] \in [3,5] \wedge [t_1,t_2] \in [1,3] \wedge [t_2,t_H] \in [2,3]$
2. ESTA and LSTA are solutions of the STP
3. Constraint Propagation

European Space Agency

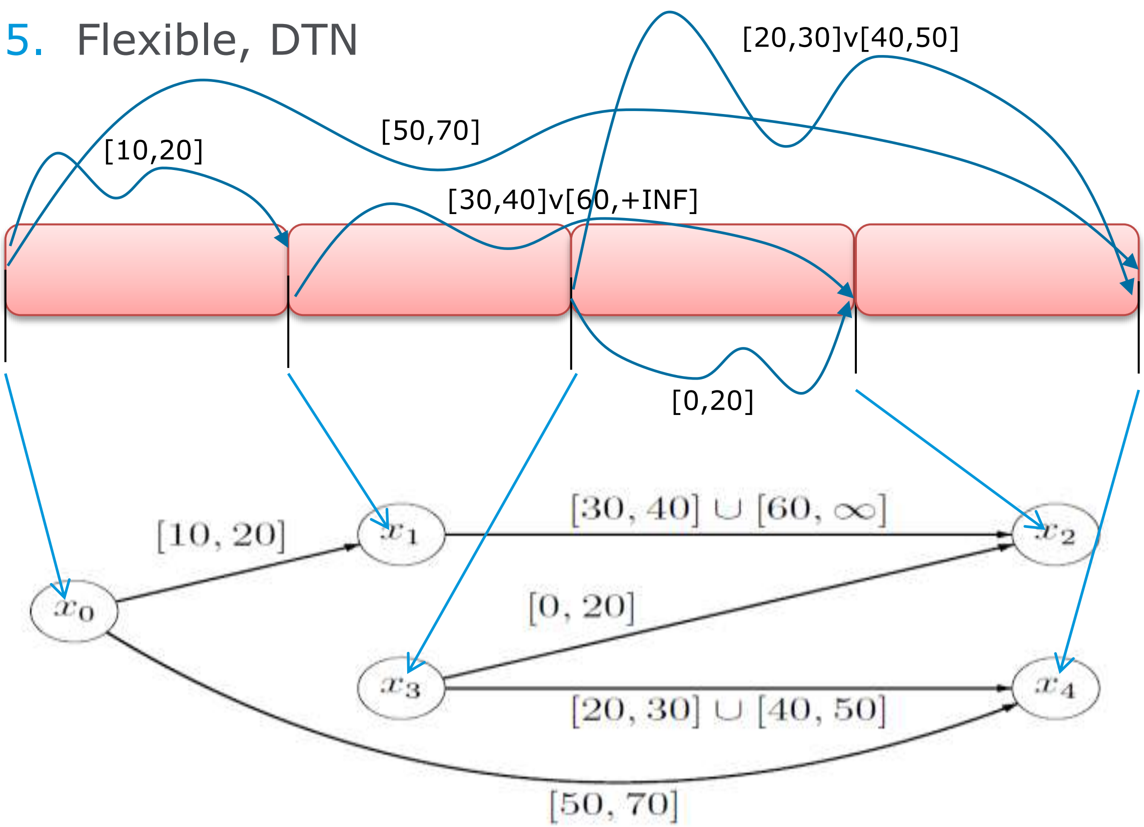
Temporal Model Classification

4. Flexible, STN generic



Temporal Model Classification

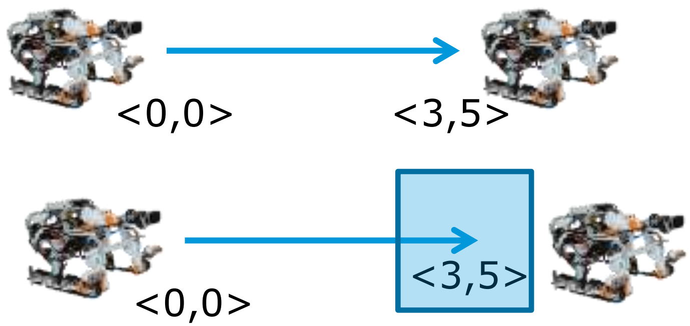
5. Flexible, DTN



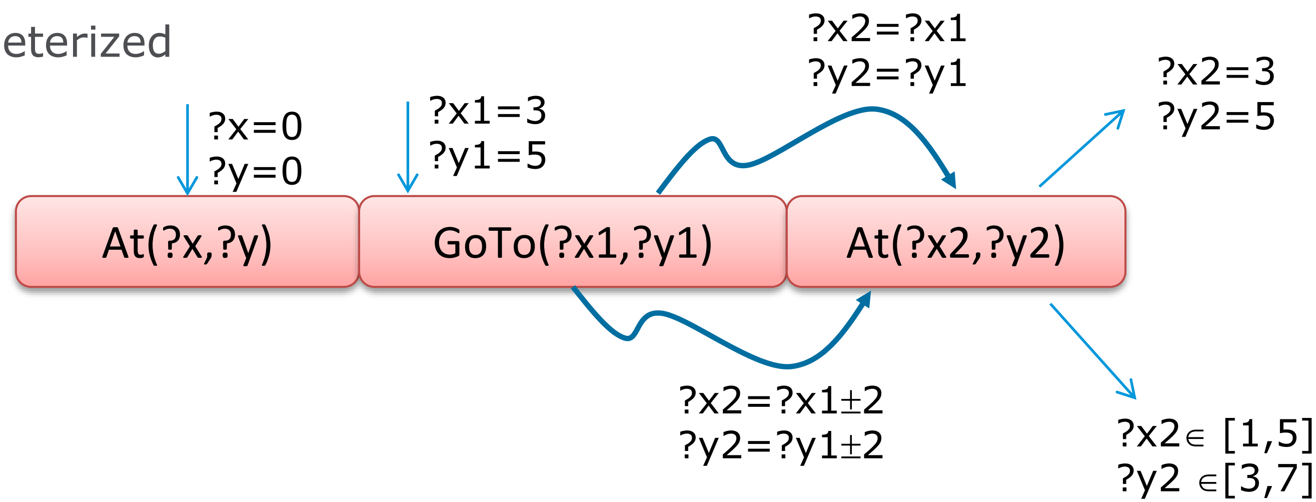
Data Model Classification

1. Ground Timeline

At(0,0) | GoTo(3,5) | At(3,5)

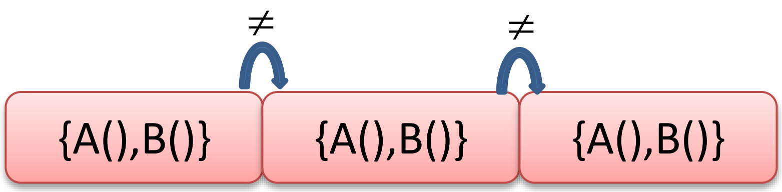


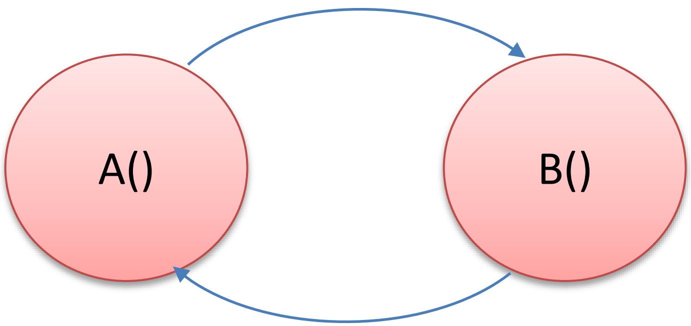
2. Parameterized

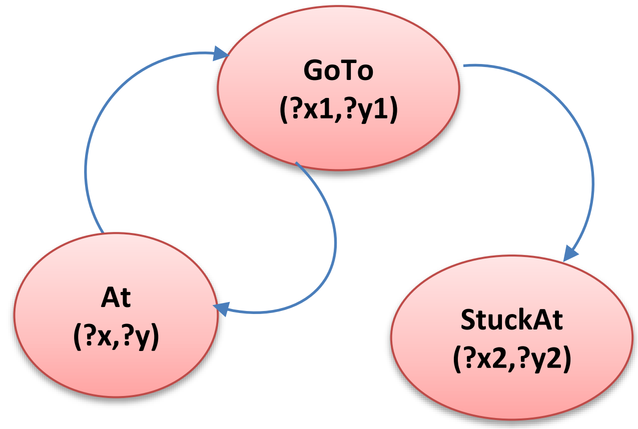


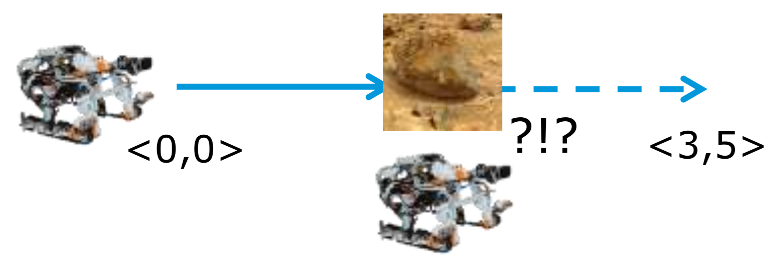
Data Model Classification

3. Multi-Valued

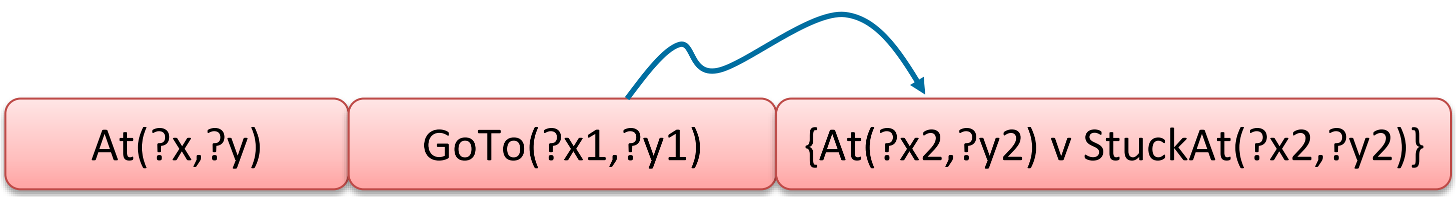




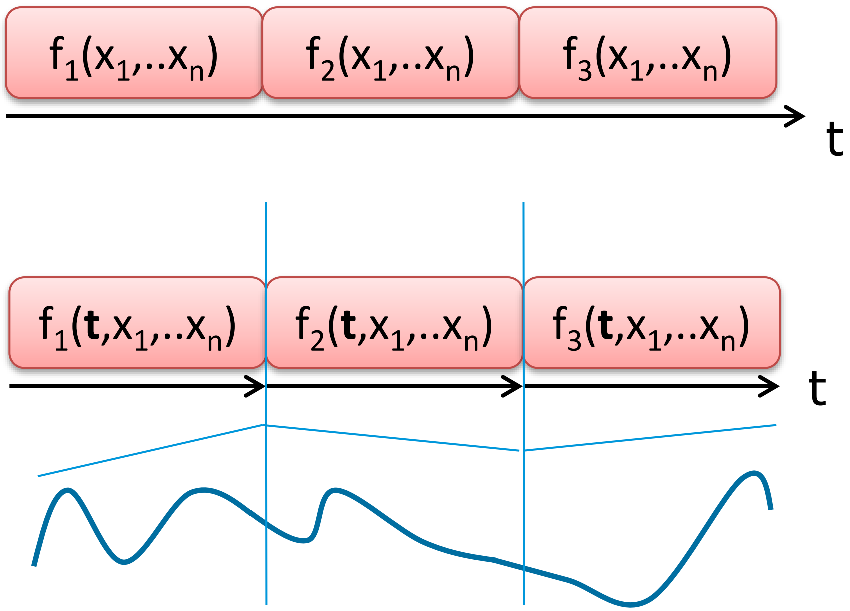




At → {?x2=?x1 ∧ ?y2=?y1}
StuckAt → {?x2≠?x1 v ?y2≠?y1}



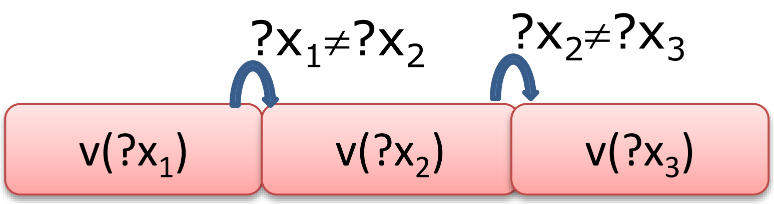
4. Generic



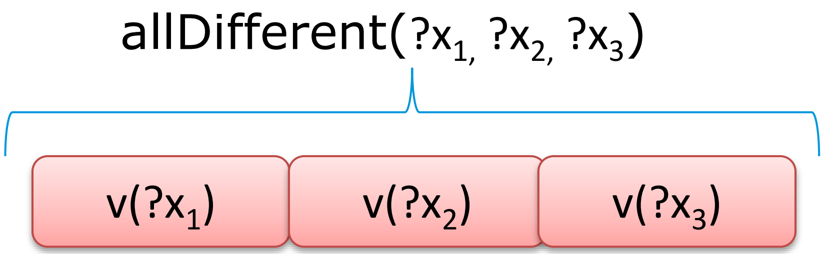
$f_1(x_1,..x_n)$
$f_2(x_1,..x_n)$
$f_3(x_1,..x_n)$
t
$f_1(\mathbf{t},x_1,..x_n)$
$f_2(\mathbf{t},x_1,..x_n)$
$f_3(\mathbf{t},x_1,..x_n)$
t

Transition Model

1. Markovian



2. Global

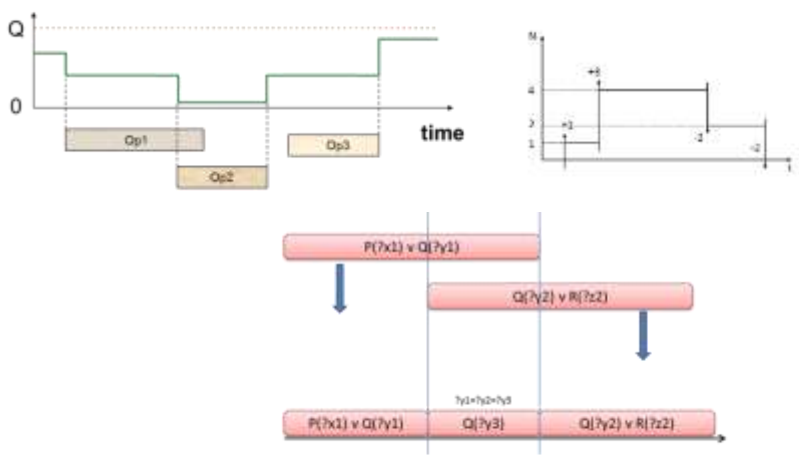


Working with Timelines

1. Assert (value or time)
    a. V = b in [2,3]
    b. tx = 3

2. Simple Constraint (on value or time)
    a. V = {b1 v b2} in [2,3]
    b. Activity
    c. Prod/cons

3. Complex Constraint
    a. Ub in [l,u]
    b. not a in [l,u]
    c. …



The “Token Principle”

No distinction needs to be made between representational primitives for *actions* and *states*. A single representational primitive, the “**token**” is sufficient to describe the evolution of system state variables over time.

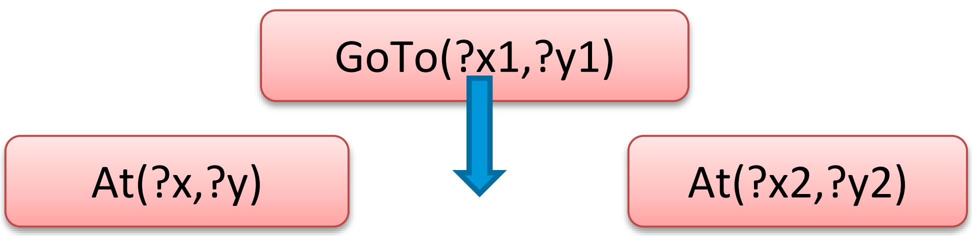
Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote agent: to go boldly where no AI system has gone before. *Artificial Intelligence*, 103(1–2):5–48.

Oh my god…
… where are the actions?!?

The “Token Principle”

No distinction needs to be made between representational primitives for *actions* and *states*. A single representational primitive, the “**token**” is sufficient to describe the evolution of system state variables over time.

Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote agent: to go boldly where no AI system has gone before. *Artificial Intelligence*, 103(1–2):5–48.



The "Token Principle"

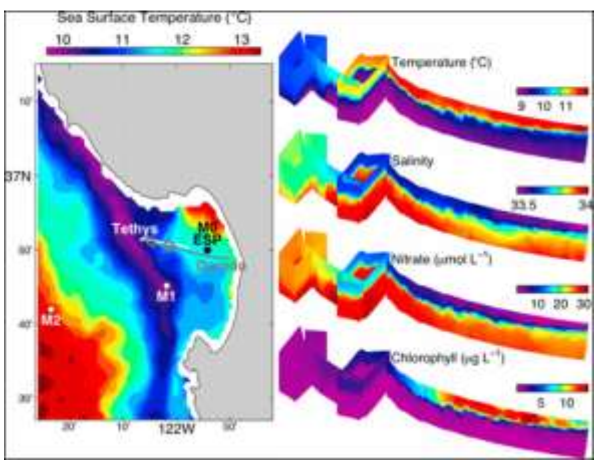
No distinction needs to be made between representational primitives for *actions* and *states*. A single representational primitive, the "**token**" is sufficient to describe the evolution of system state variables over time.

Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote agent: to go boldly where no AI system has gone before. *Artificial Intelligence*, 103(1–2):5–48.

AscendDescend(?d2)

Hoovering(?d)

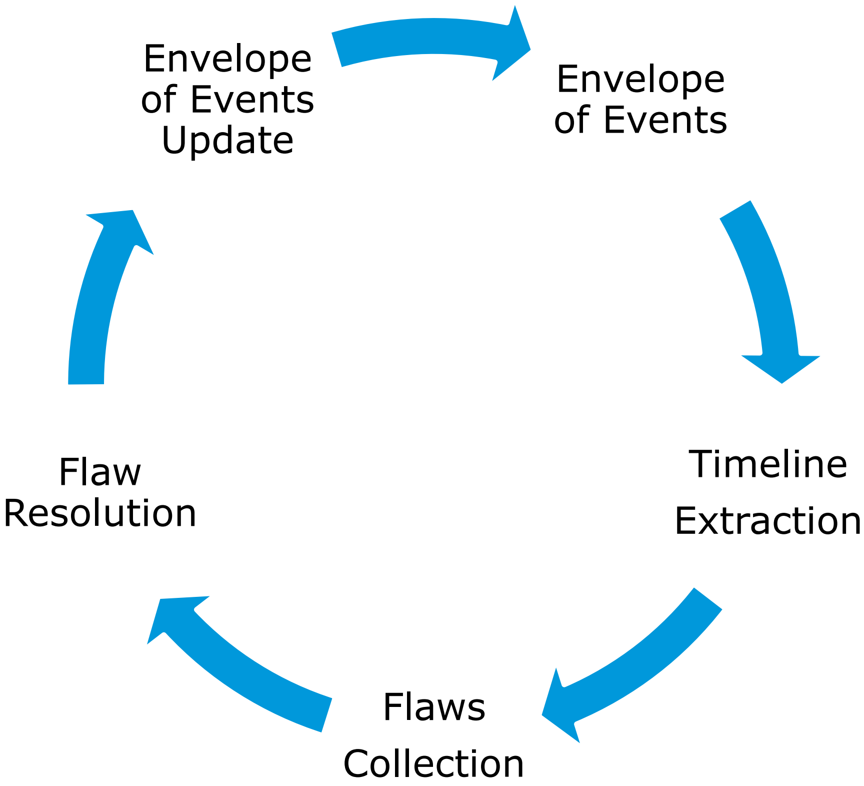
Hoovering(?d2)



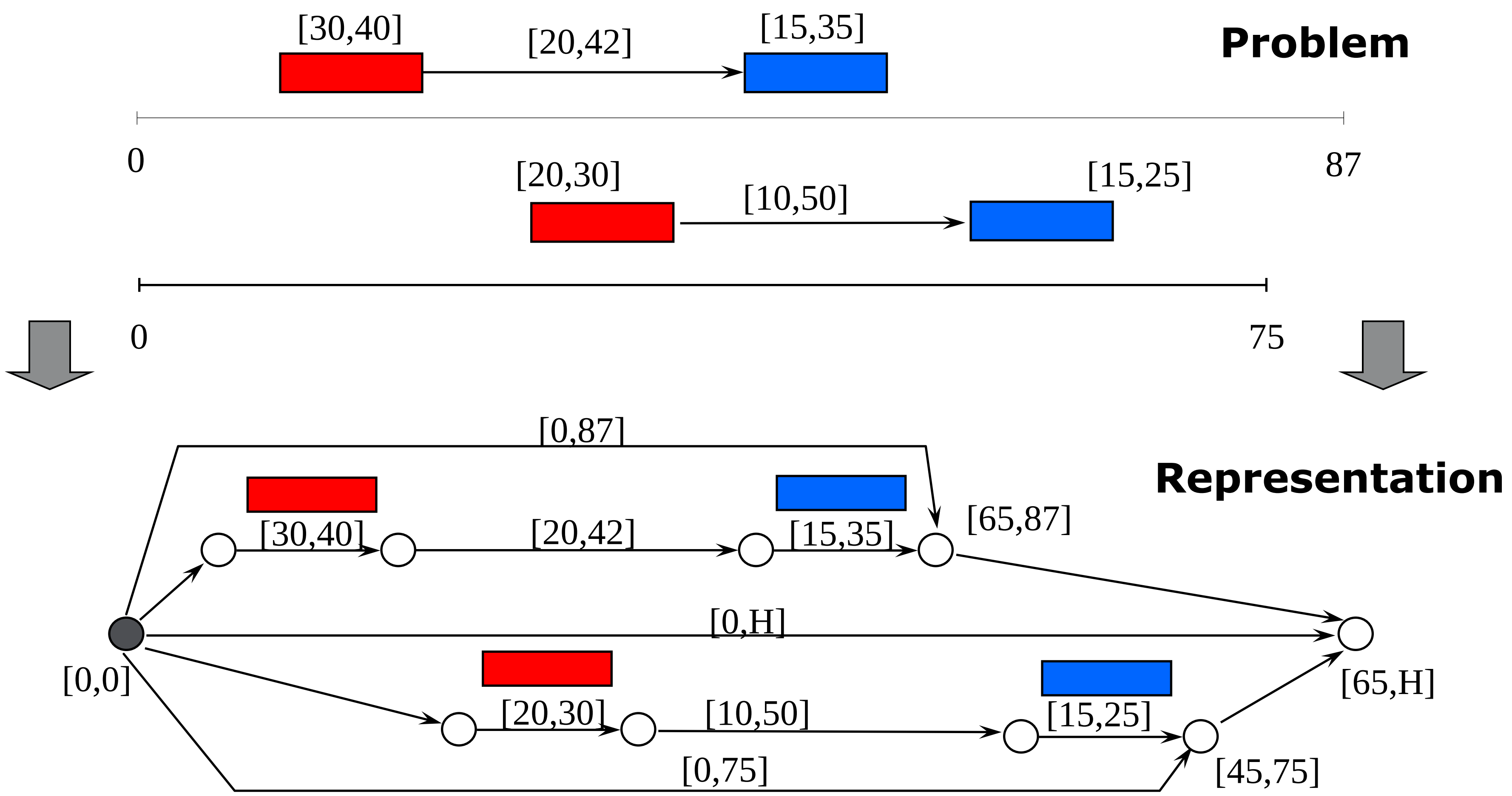
Problem Solving with Timelines:

- Timeline Extraction Procedures
- Flaws Collection
- Flaws Resolution

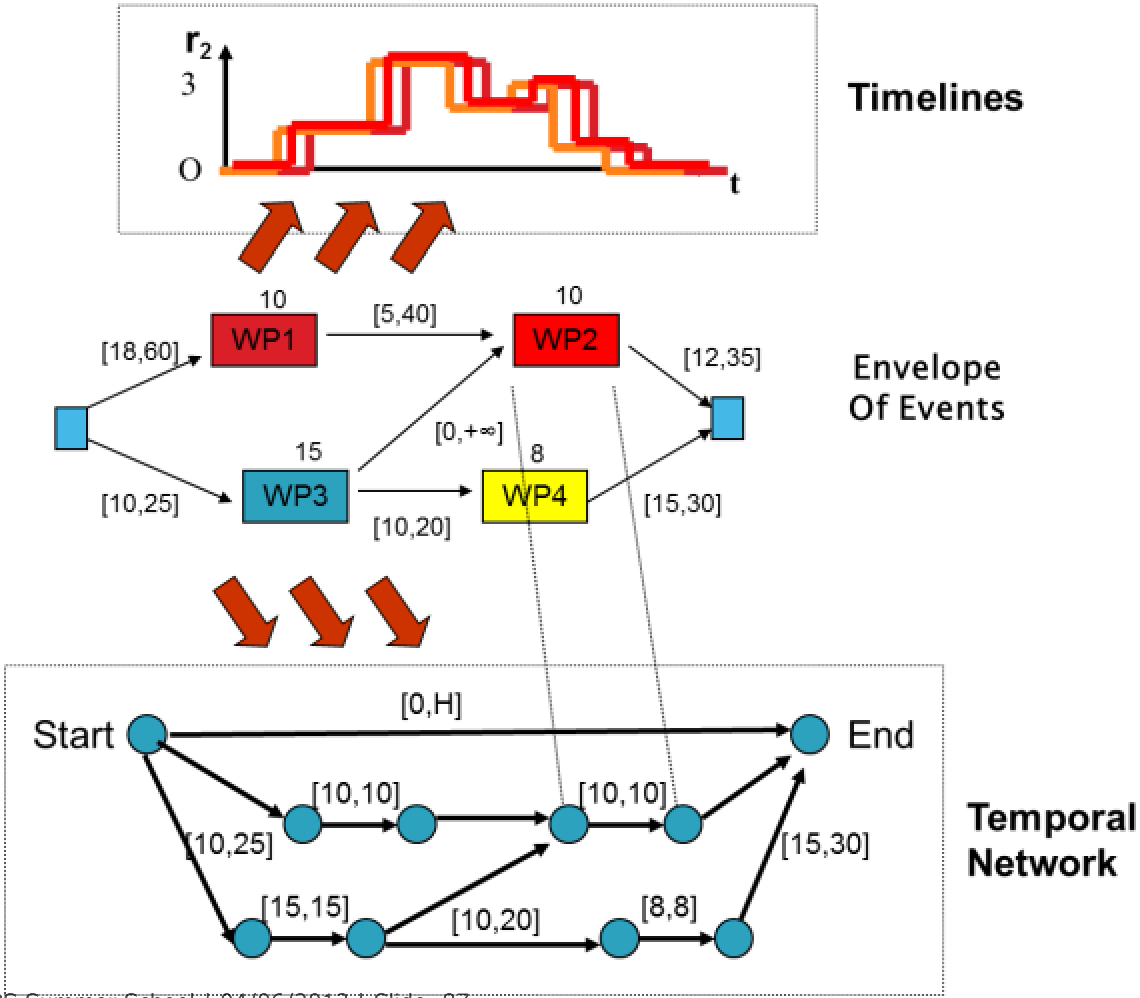
Problem Solving Main Steps



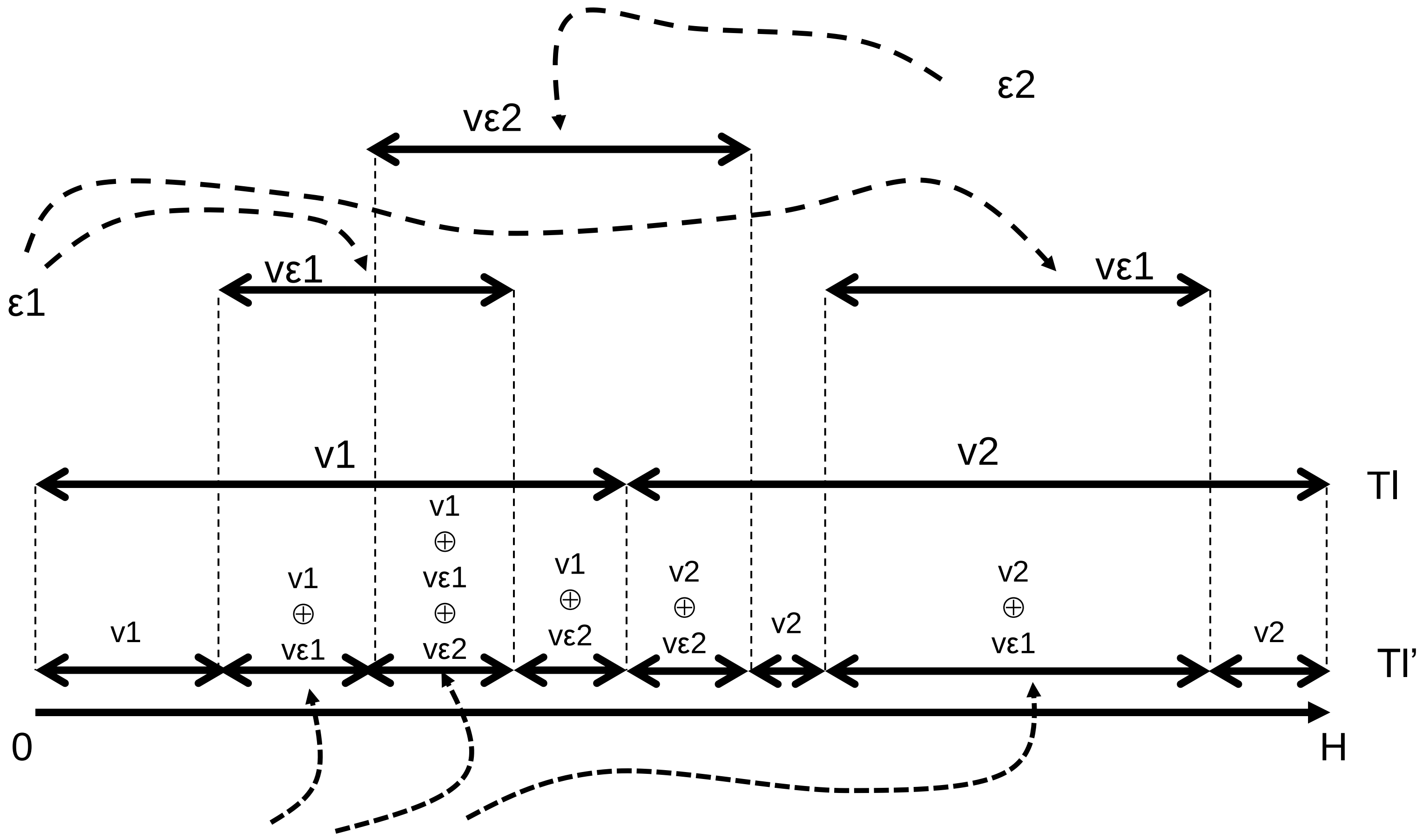
Layered Structure



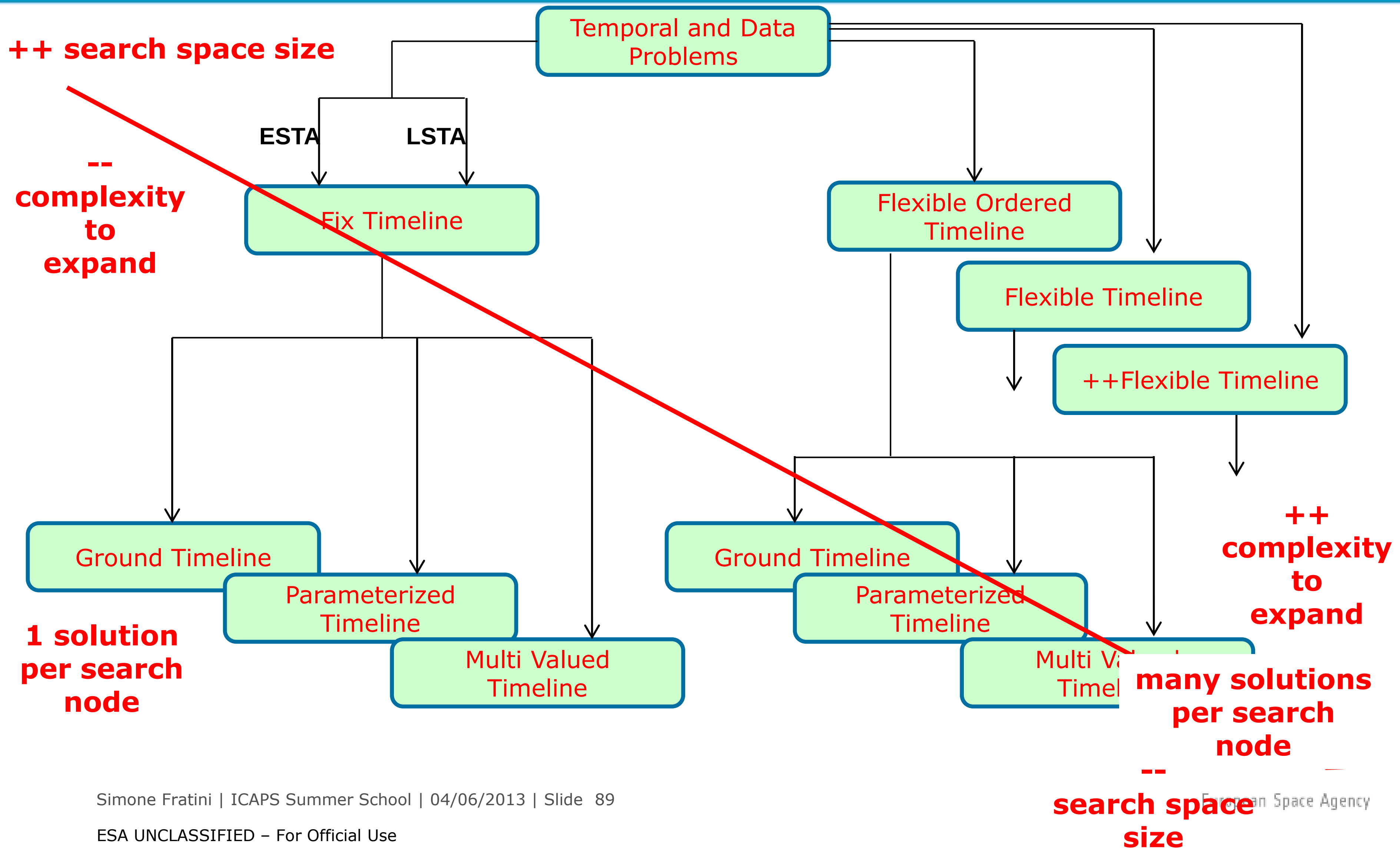
Timeline Extraction Procedure



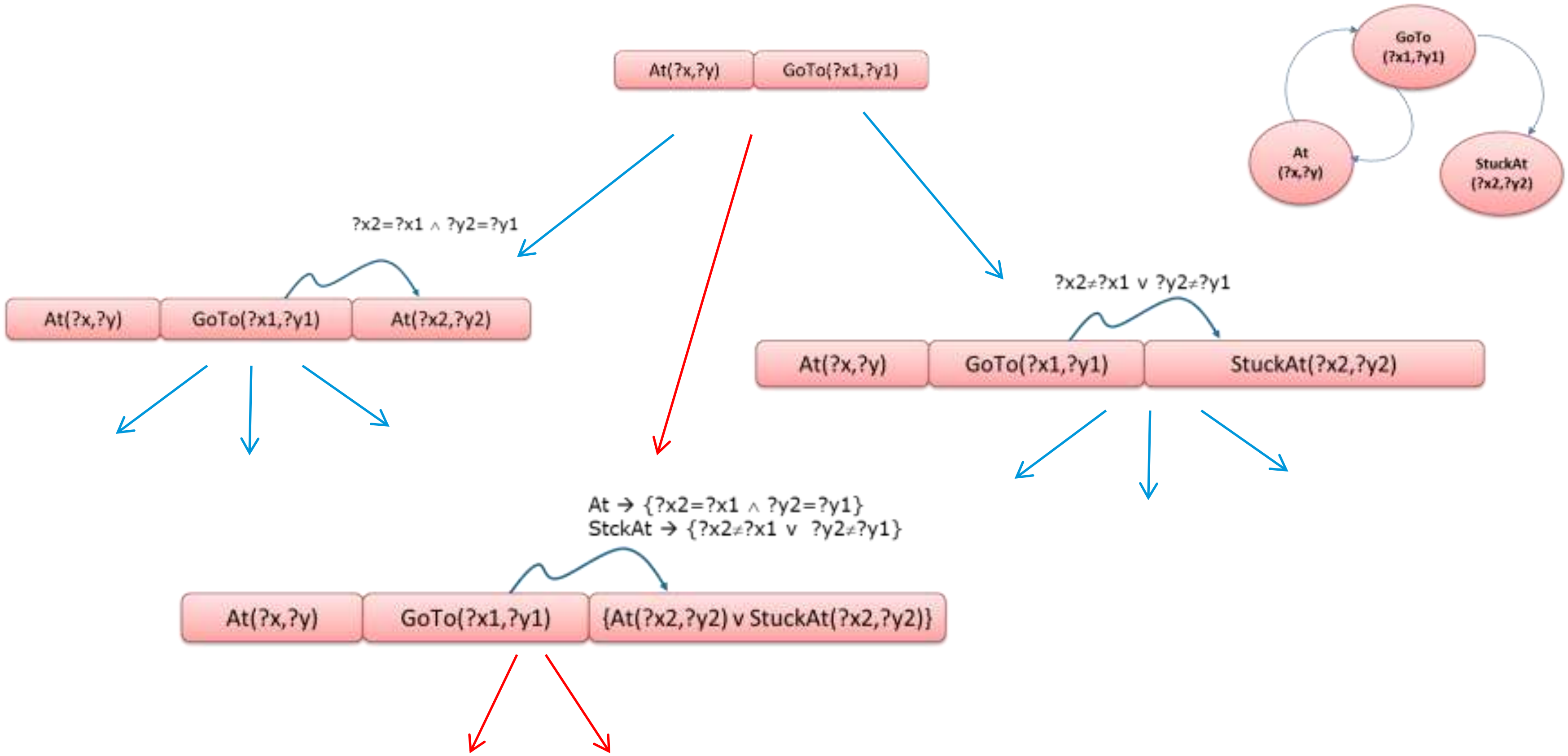
Temporal Fragmentation



Timeline Extraction Procedure



Timeline Extraction Procedure



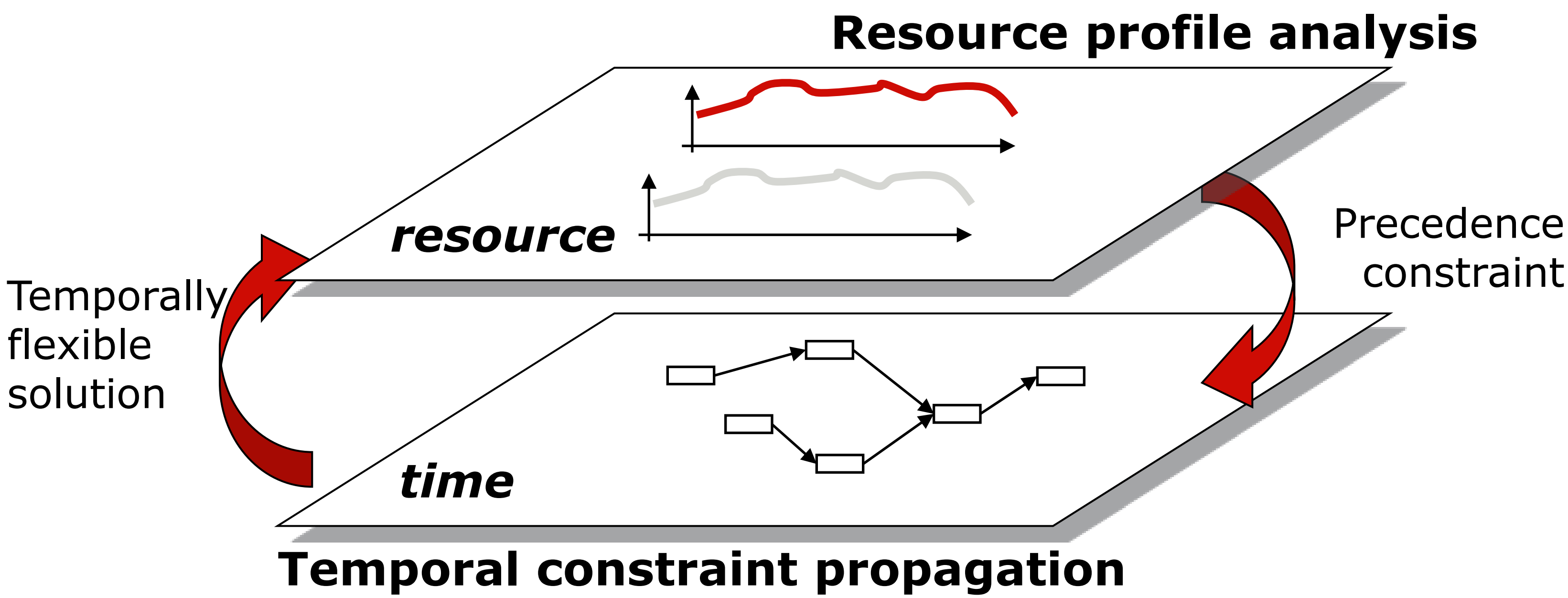
Flaws

1. Open Condition Flaws
   - Expand (add tokens)
   - Unify (add constraints)

2. Ordering Flaws (Conflicts on Timelines)
   - **PCP (add temporal precedence constraints)**

3. Unbound Variable Flaws
   - Add Value Constraint

Precedence Constraint Posting

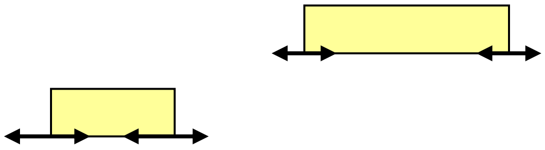


1. Variables:
   a. start and end times of each activity
2. Domains
   a. A schedule horizon [0, H]
3. Constraints
   a. Temporal constraints (e.g., duration of activities, min-max separation between activities, simple precedences)
   b. Resource constraints (e.g., bounds on capacities)

State Variable Scheduling

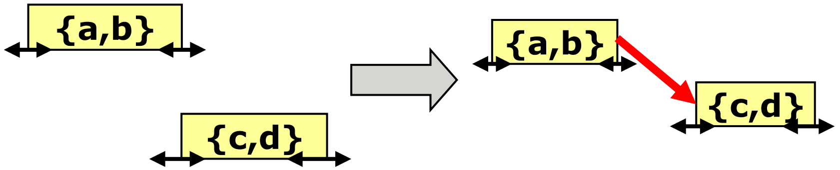
Surely not overlapping

- No action



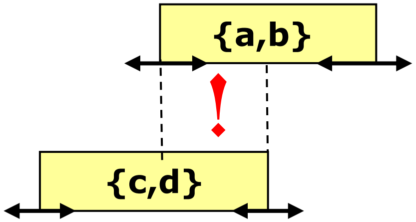
Could overlap start or end

- Propagation



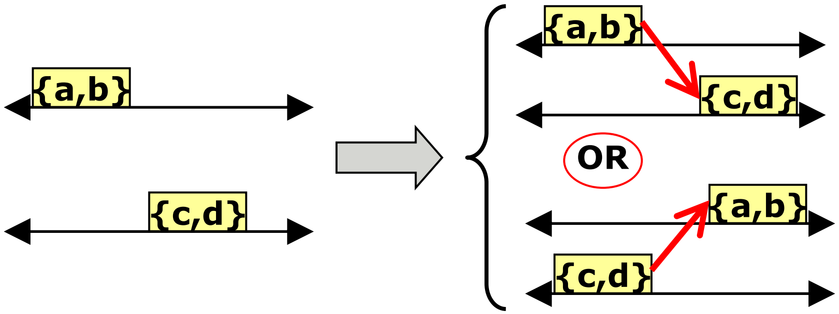
Surely overlapping

- Evaluate intersection



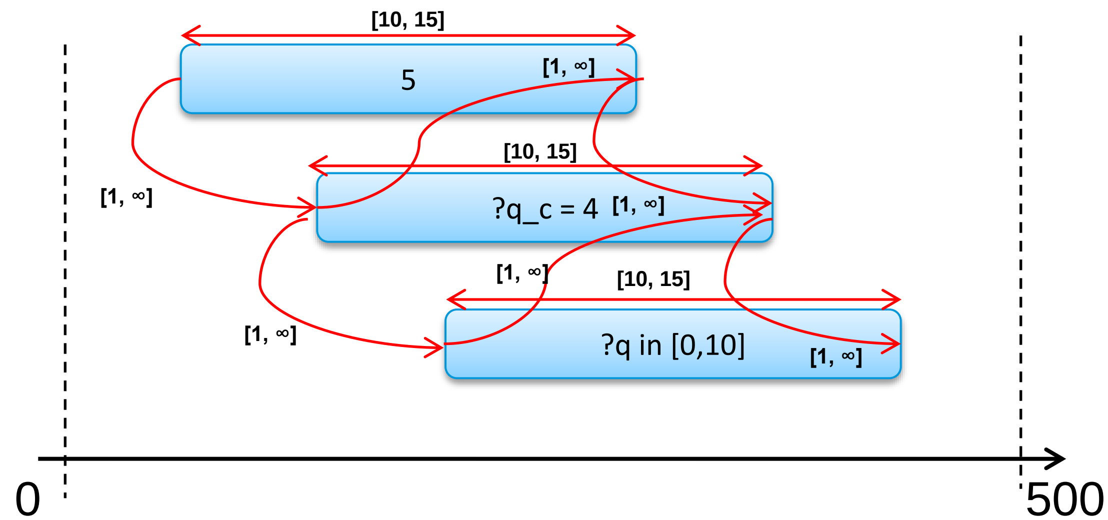
Both feasible orderings

- Scheduling (Search)

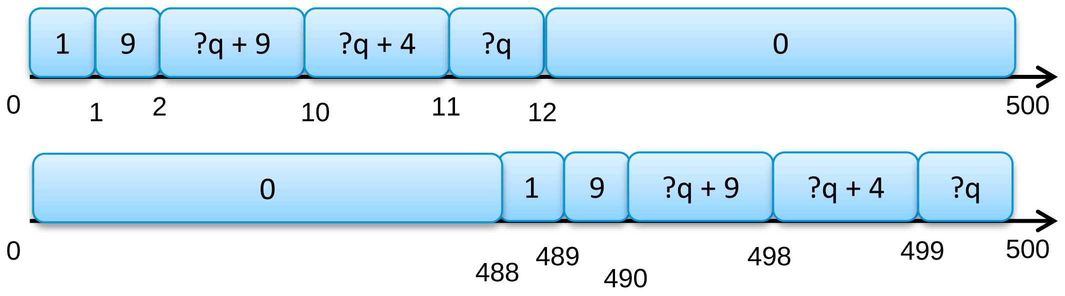


Example

1. The Problem:



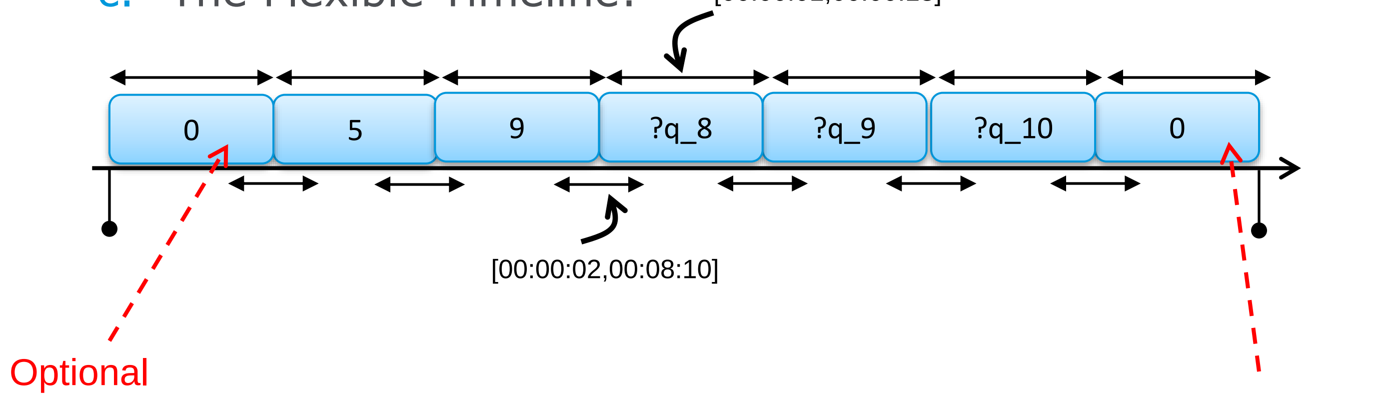
2. The Timelines:
   a. ESTA
   b. LSTA



Example

$[0,3] = 5;$
$[1,3] = ?q\_c = 4;$
$[2,3] = ?q \text{ in } [1,10]$

c. The Flexible Timeline:



$?q\_8 = ?q + ?q\_c + 5$
$?q\_9 = ?q + ?q\_c$
$?q\_10 = ?q$

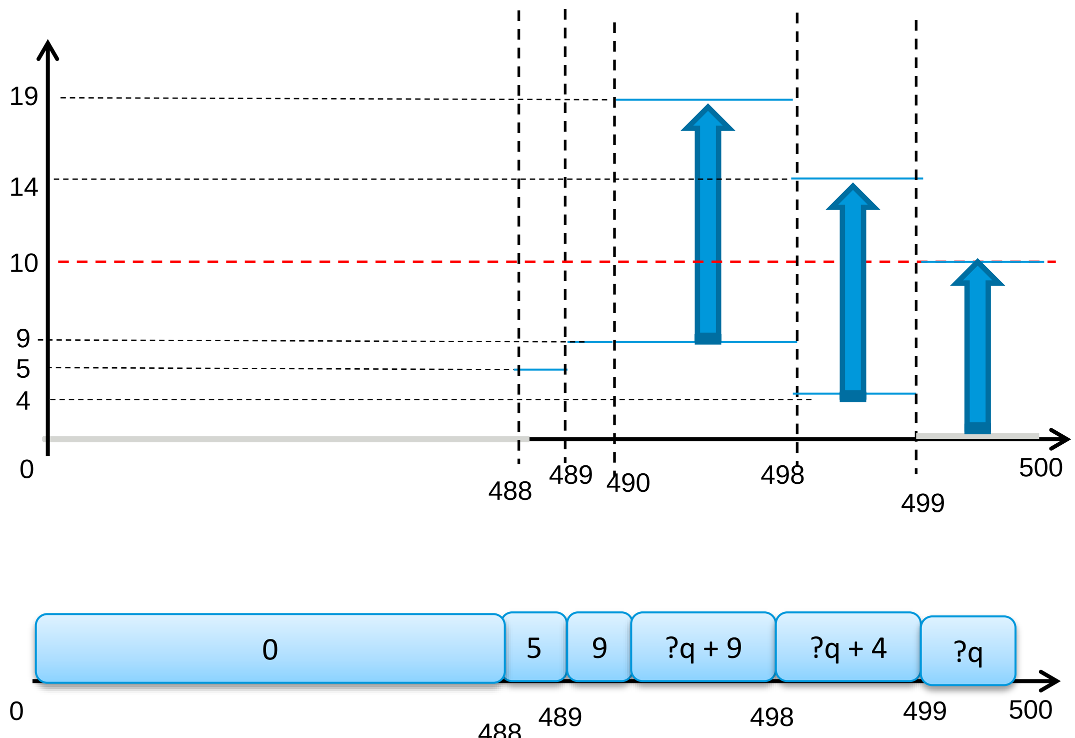
Time-Flexible Timeline: ReusableResource.Timeline #7 – FLEXIBLE

TPId	Start [lb,ub]	Duration [lb,ub]	Value	Decisions
[0]	[00:00:00,00:00:00]	[00:00:00,00:08:08] (O)	0	
[3]	[00:00:00,00:08:08]	[00:00:01,00:00:13]	5	<[0,3]>
[6]	[00:00:01,00:08:09]	[00:00:01,00:00:13]	9	<[0,3],[1,3]>
[9]	[00:00:02,00:08:10]	[00:00:01,00:00:13]	?q_8	<[0,3],[1,3],[2,3]>
[4]	[00:00:10,00:08:18]	[00:00:01,00:00:13]	?q_9	<[1,3],[2,3]>
[7]	[00:00:11,00:08:19]	[00:00:01,00:00:13]	?q_10	<[2,3]>
[10]	[00:00:12,00:08:20]	[00:00:00,00:08:08] (O)	0	
[1]	[00:08:20,00:08:20]			

Example

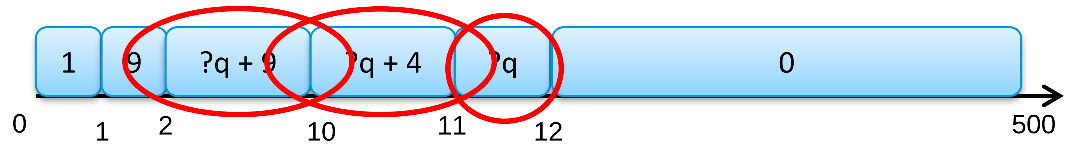
3. The Envelopes of Profiles:

   a. LSTA:



Example

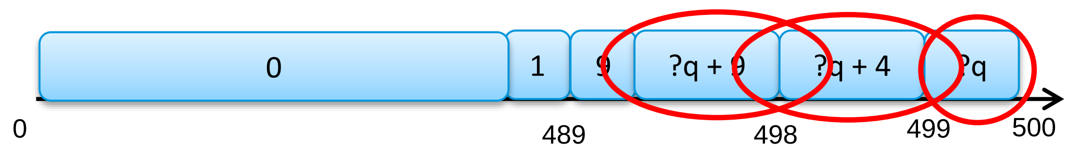
4. Reasoning on Timelines (Problem Solving):



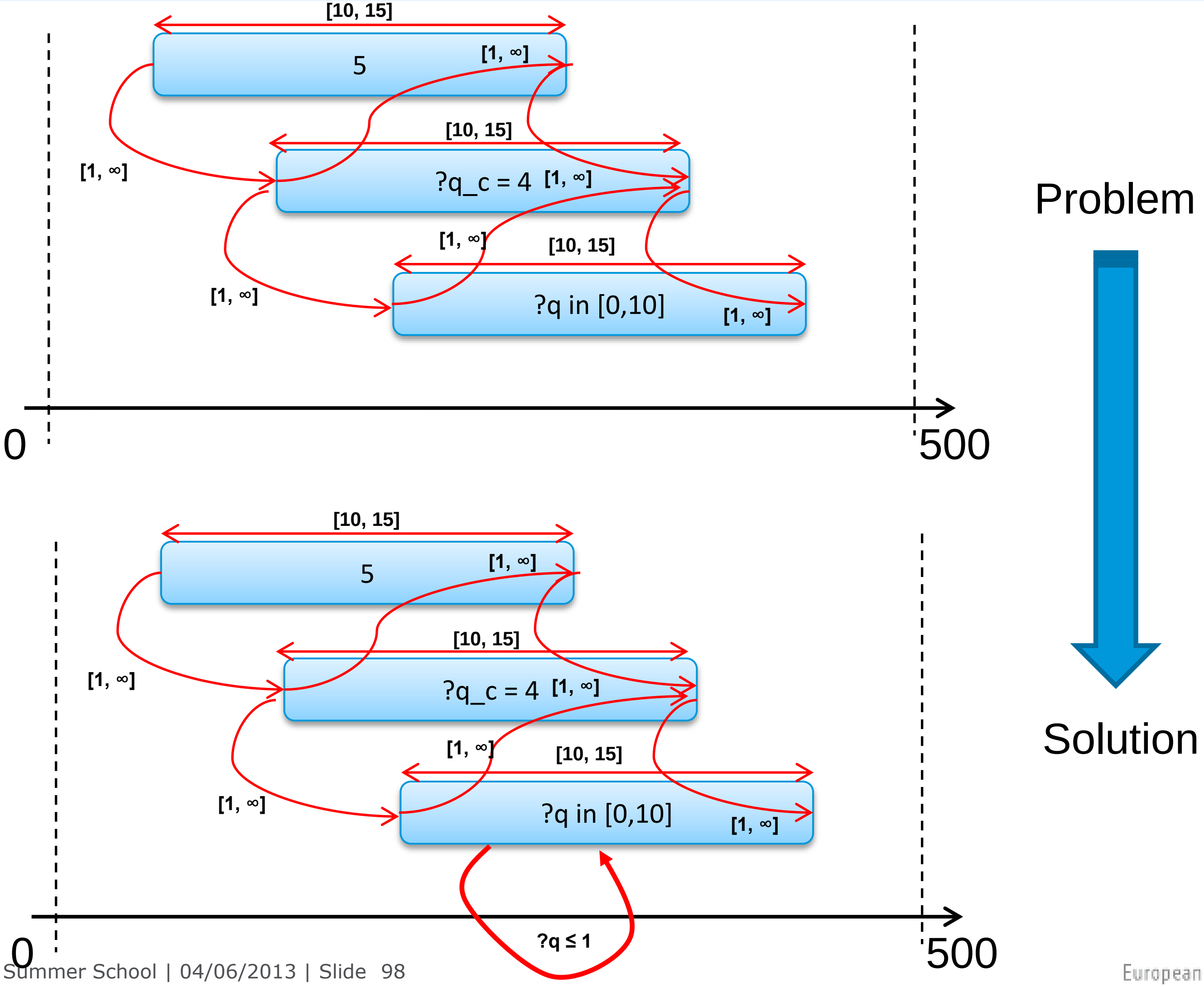
Max_Cap = 10

?q+9 ≤ 10
?q+4 ≤ 10
?q ≤ 10

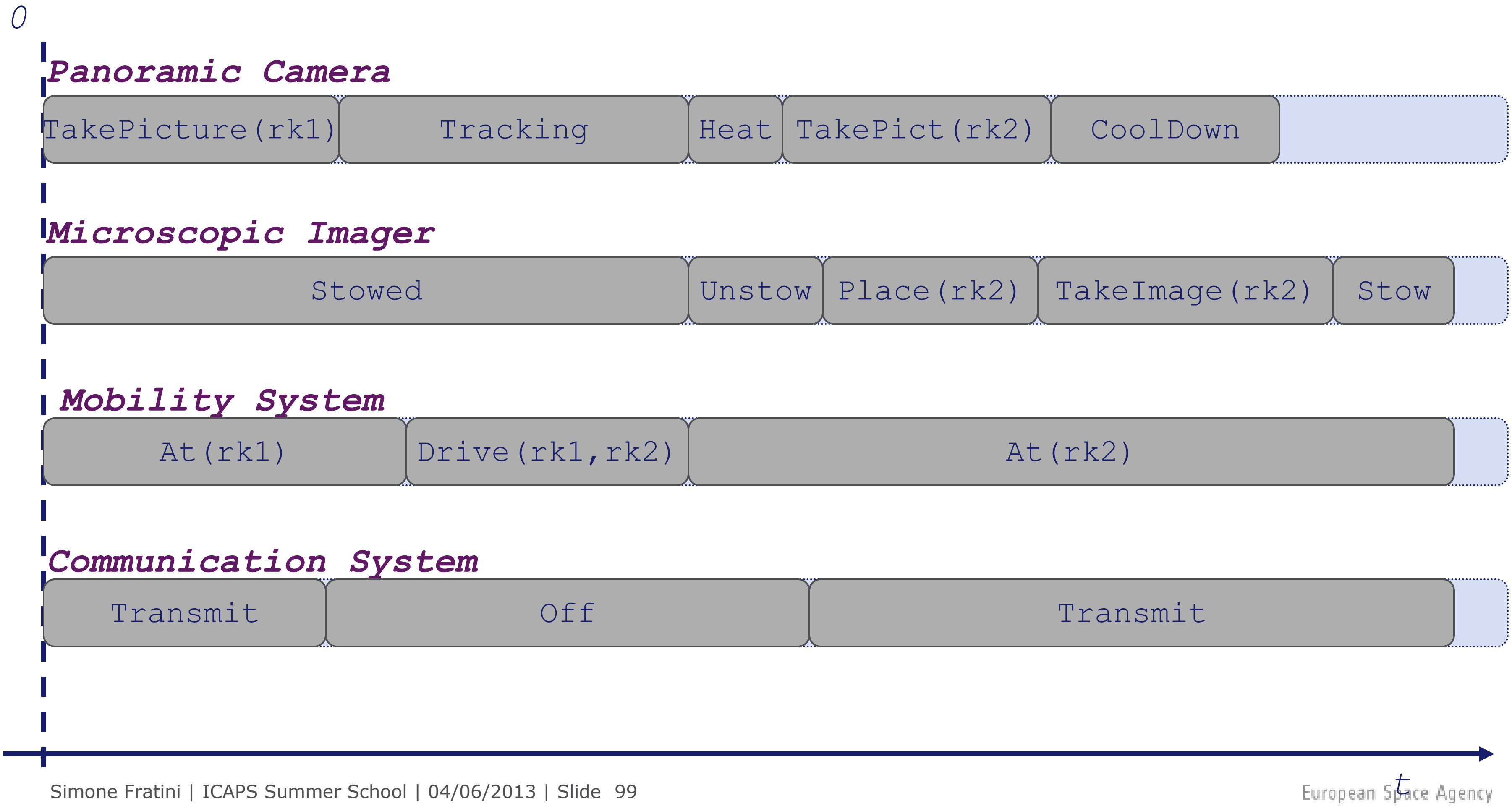
?q in [0,1]



Example


[10, 15]
5
[1, ∞]
[10, 15]
?q_c = 4
[1, ∞]
[1, ∞]
[1, ∞]
[10, 15]
[1, ∞]
?q in [0,10]
[1, ∞]
0
500
Problem
Solution
[10, 15]
5
[1, ∞]
[10, 15]
?q_c = 4
[1, ∞]
[1, ∞]
[1, ∞]
[10, 15]
[1, ∞]
?q in [0,10]
[1, ∞]
?q ≤ 1
0
500

One more example

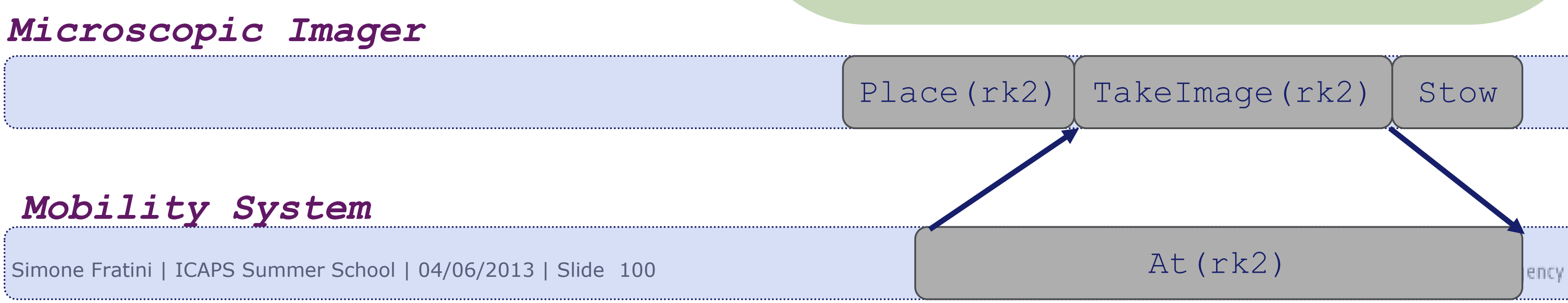


Courtesy of S. Bernardini

The NDDL Language

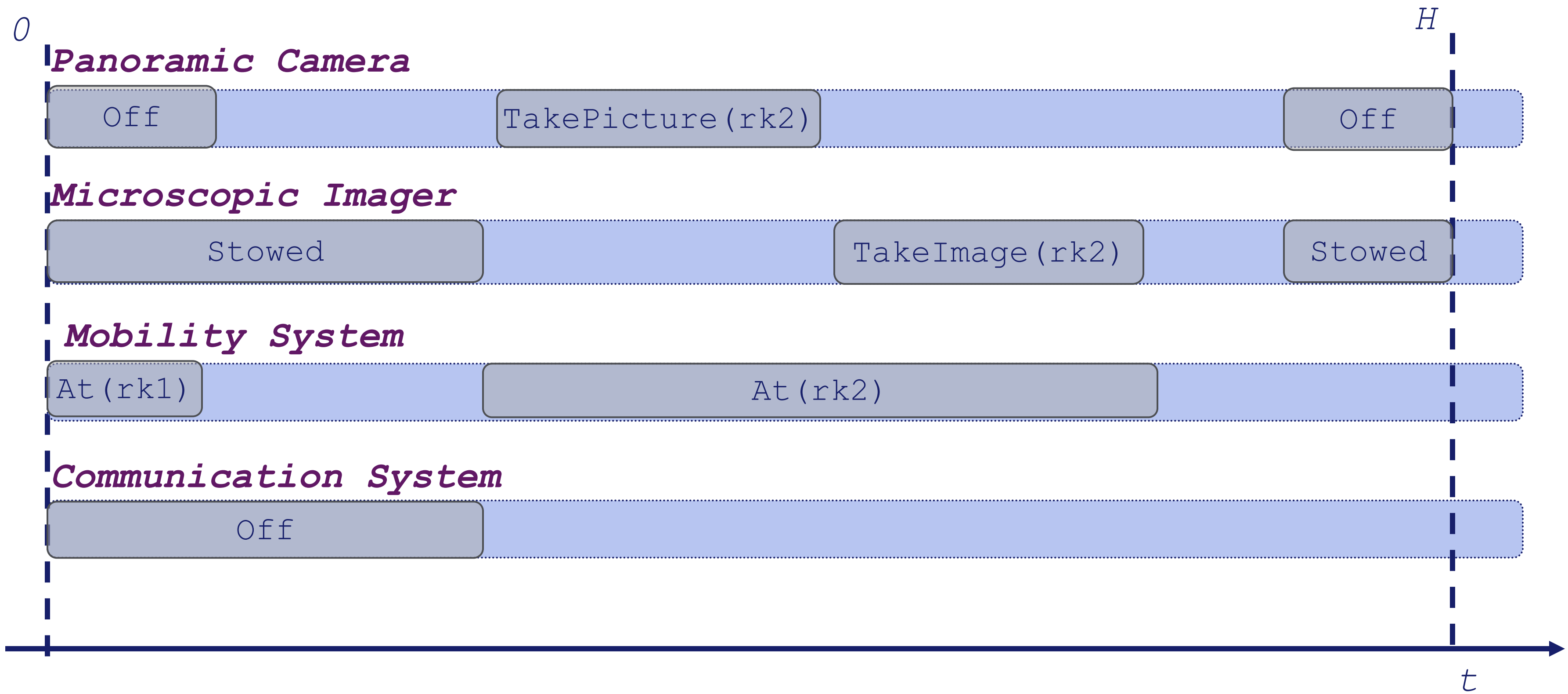
```
class MicroscopicImager{
   predicate TakeImage{
     Rock rock;
     eq(duration, [1,10]);};
   predicate Stow{
      eq(duration, [2,6]);};
   predicate Unstow{
     eq(duration, [2,6]);};
   predicate Place{
     Rock rock;
     eq(duration, [3,12]);};
   predicate Stowed;}
```

```
MicroscopicImager::TakeImage{
   met_by(Place _pl);
   eq(_pl.rock, rock);

   bool another_image;
   if (another_image == true)
     meets(Place _pl);
   if (another_image == false)
     meets(Stow _st);

   contained_by(MobilitySystem.At _at)
   eq(_at.rock, rock);

}
```



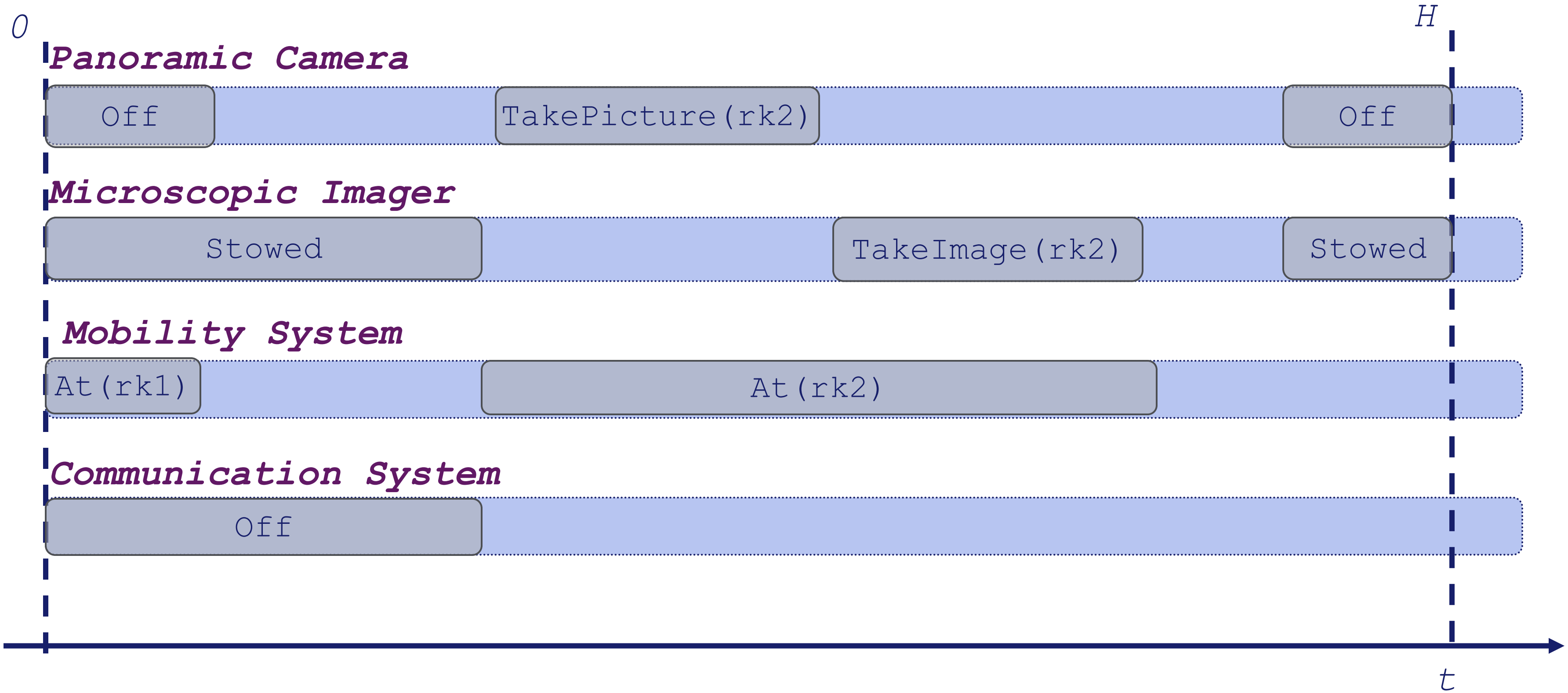
Courtesy of S. Bernardini

Planning Problem



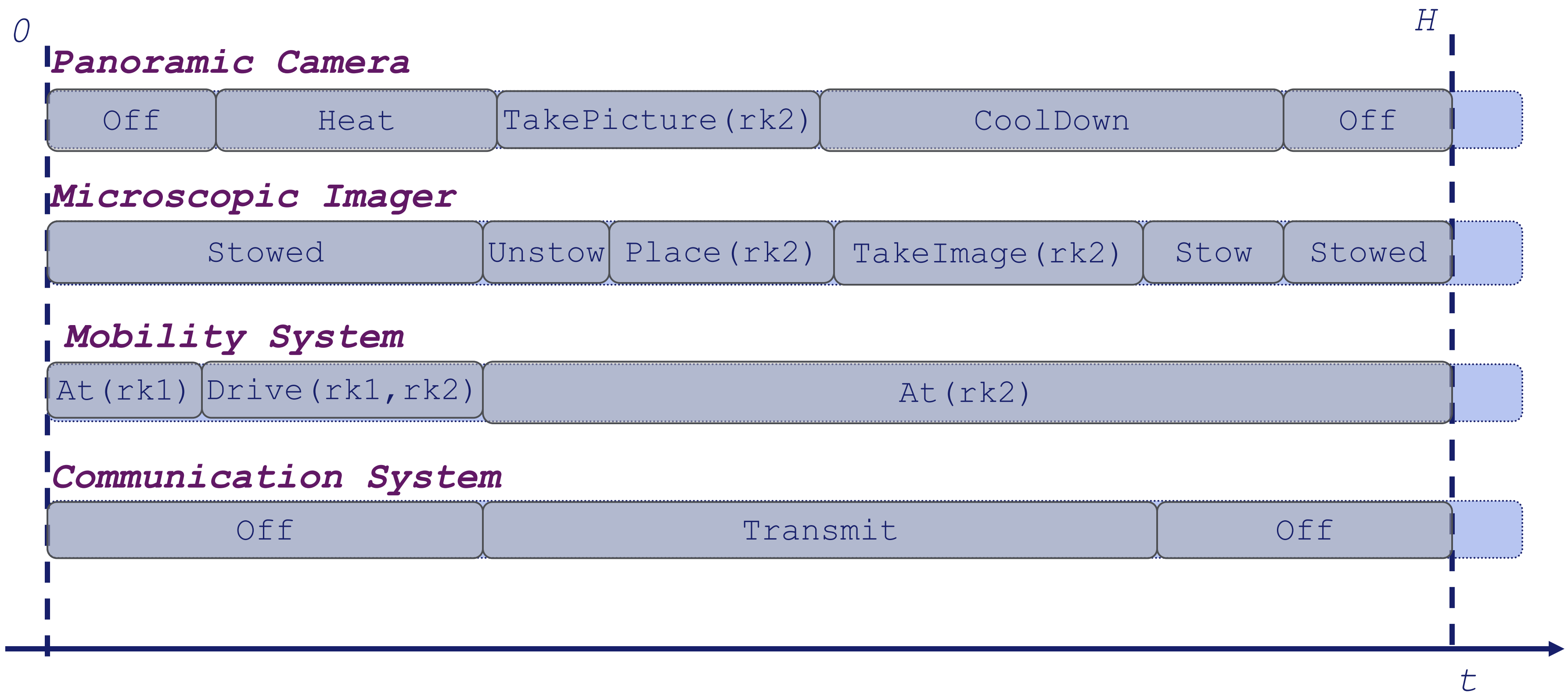
Courtesy of S. Bernardini

Solution Plan



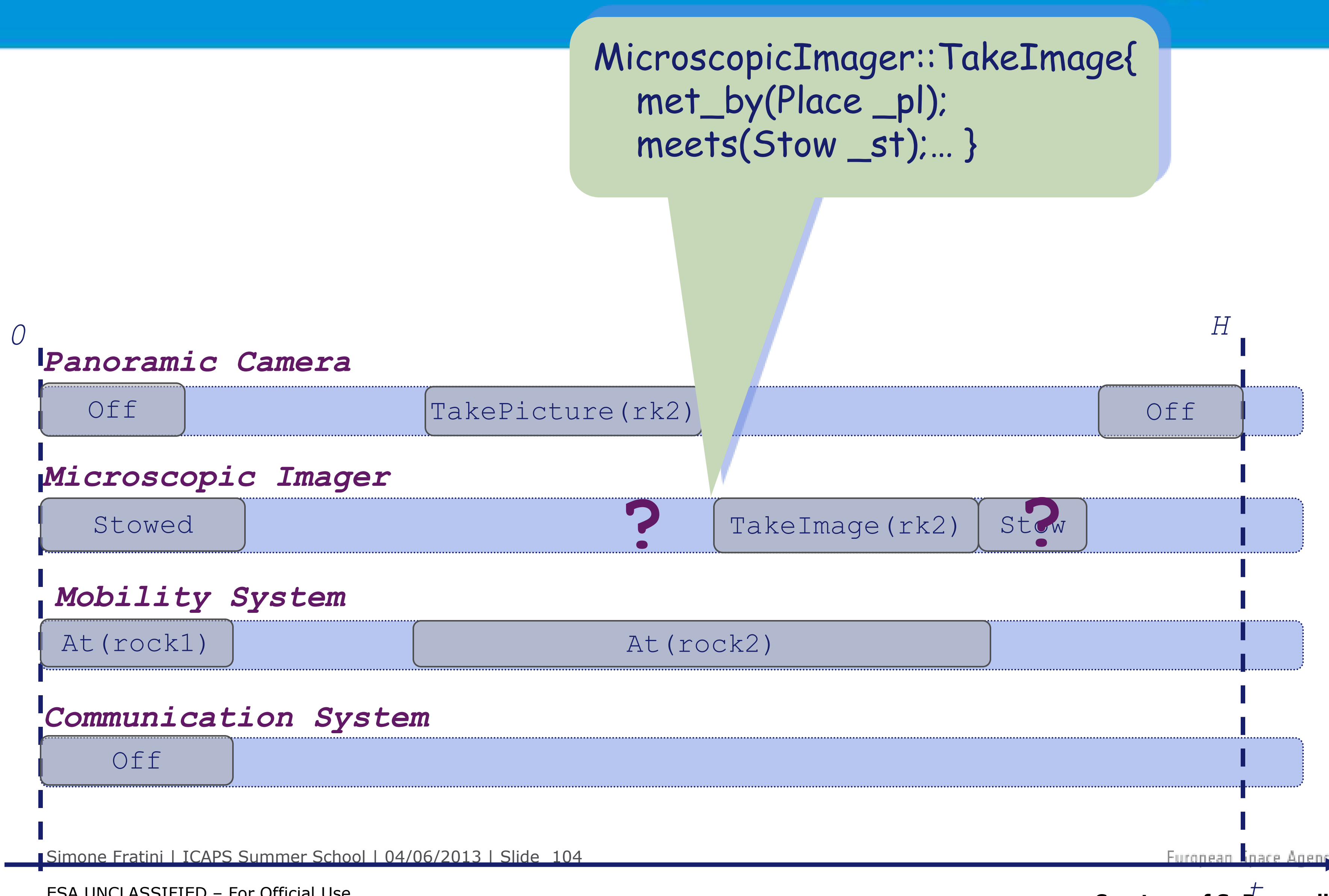
Courtesy of S. Bernardini

Solution Plan



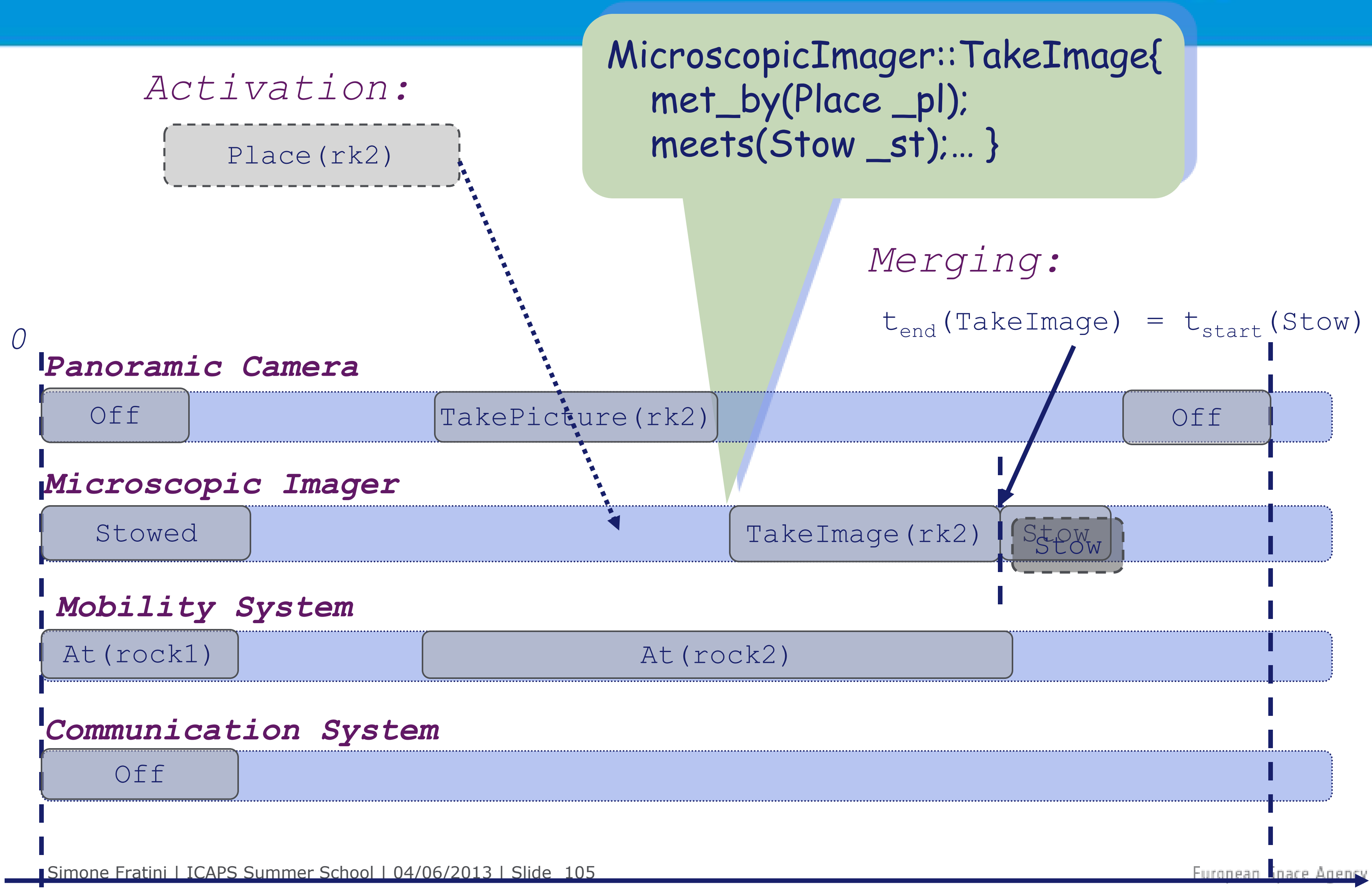
Courtesy of S. Bernardini

Open Condition Flaws



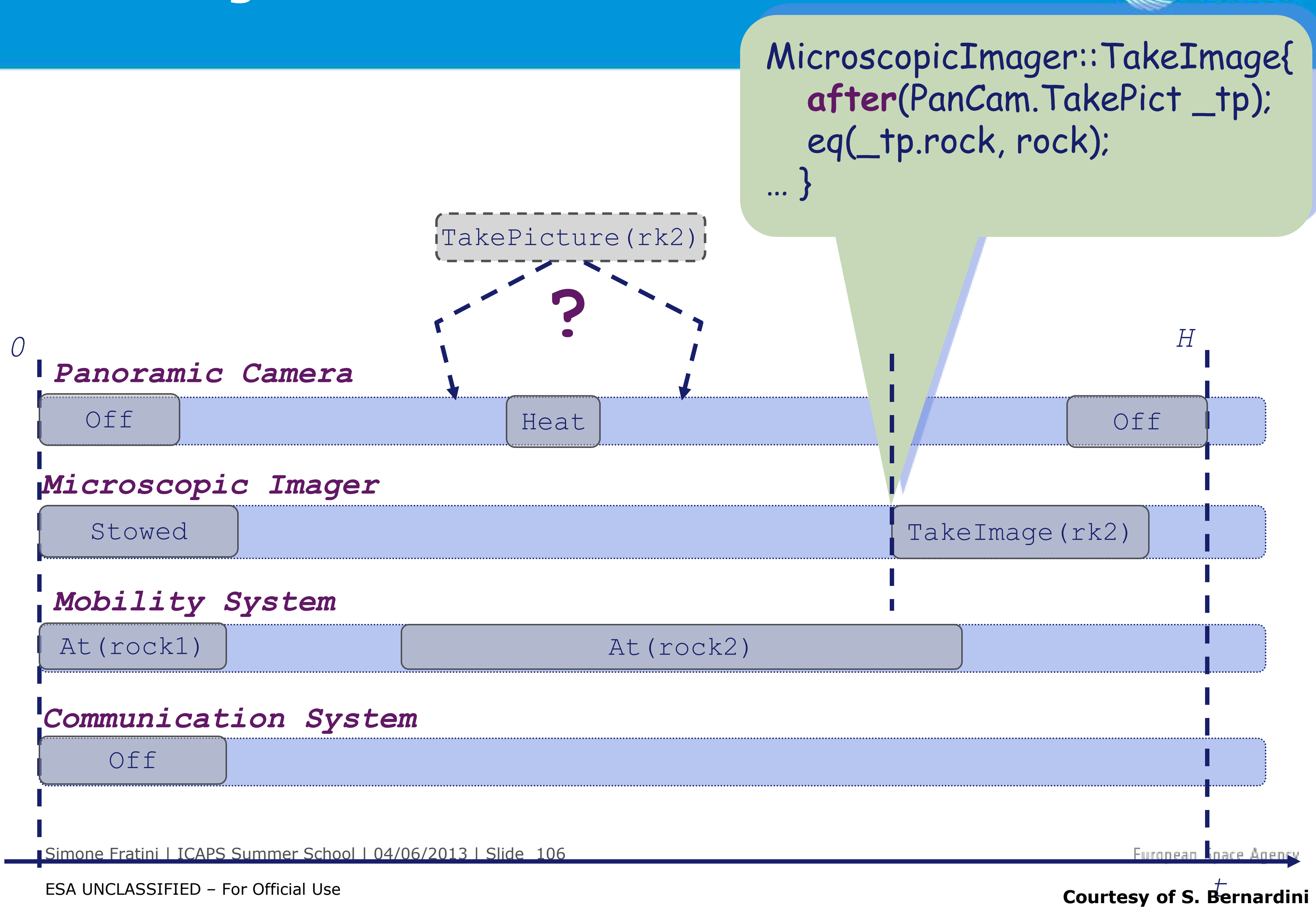
Courtesy of S. Bernardini

Open Condition Resolvers



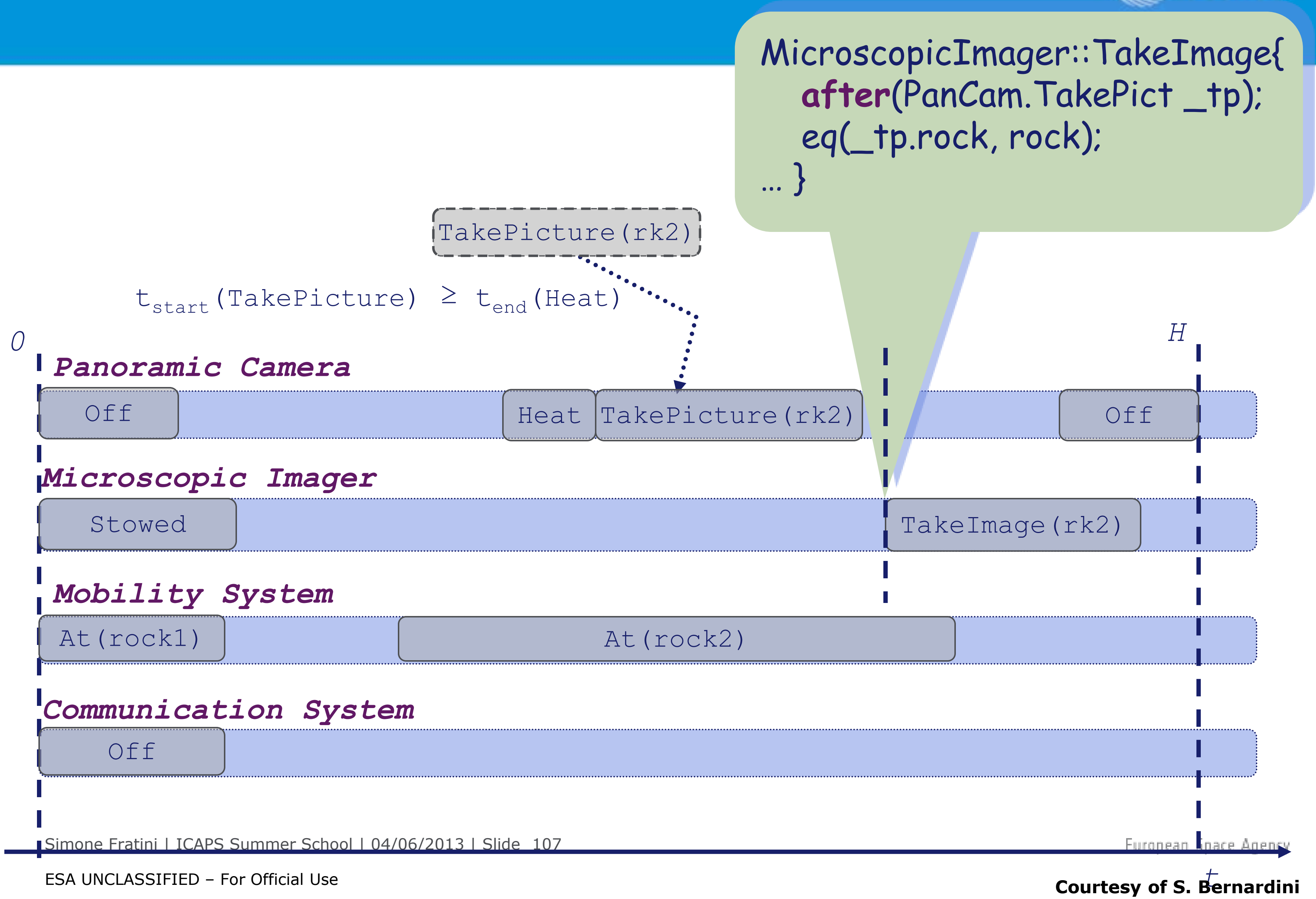
Courtesy of S. Bernardini

Ordering Flaws



Courtesy of S. Bernardini

Ordering Flaw Resolvers



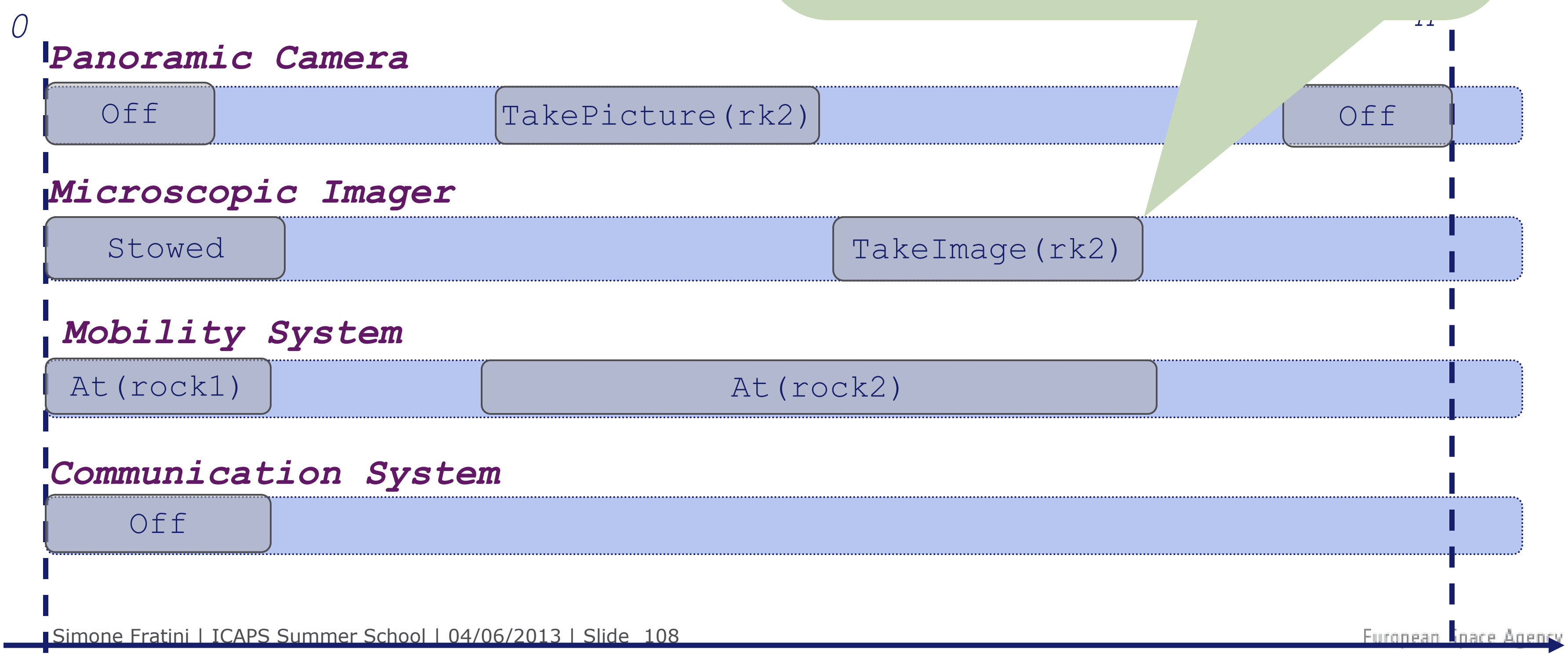
Courtesy of S. Bernardini

Unbound Variable Flaws

MicroscopicImager::TakeImage{
bool another_image;
if (**another_image == true**)
    meets(Place _pl);
  if (**another_image == false**)
    meets(Stow _st);}

?

another_image == {true, false}

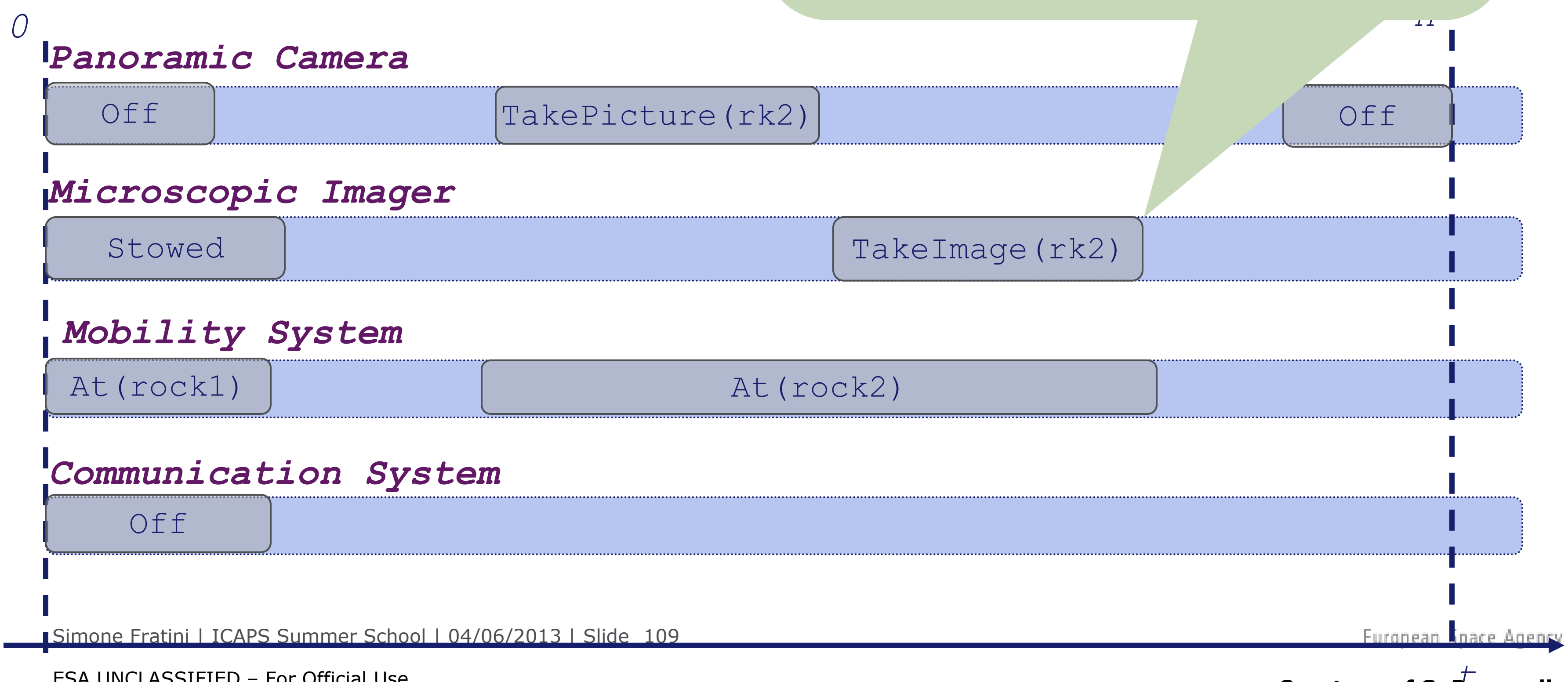


Courtesy of S. Bernardini

Unbound Variable Resolvers

another_image = true

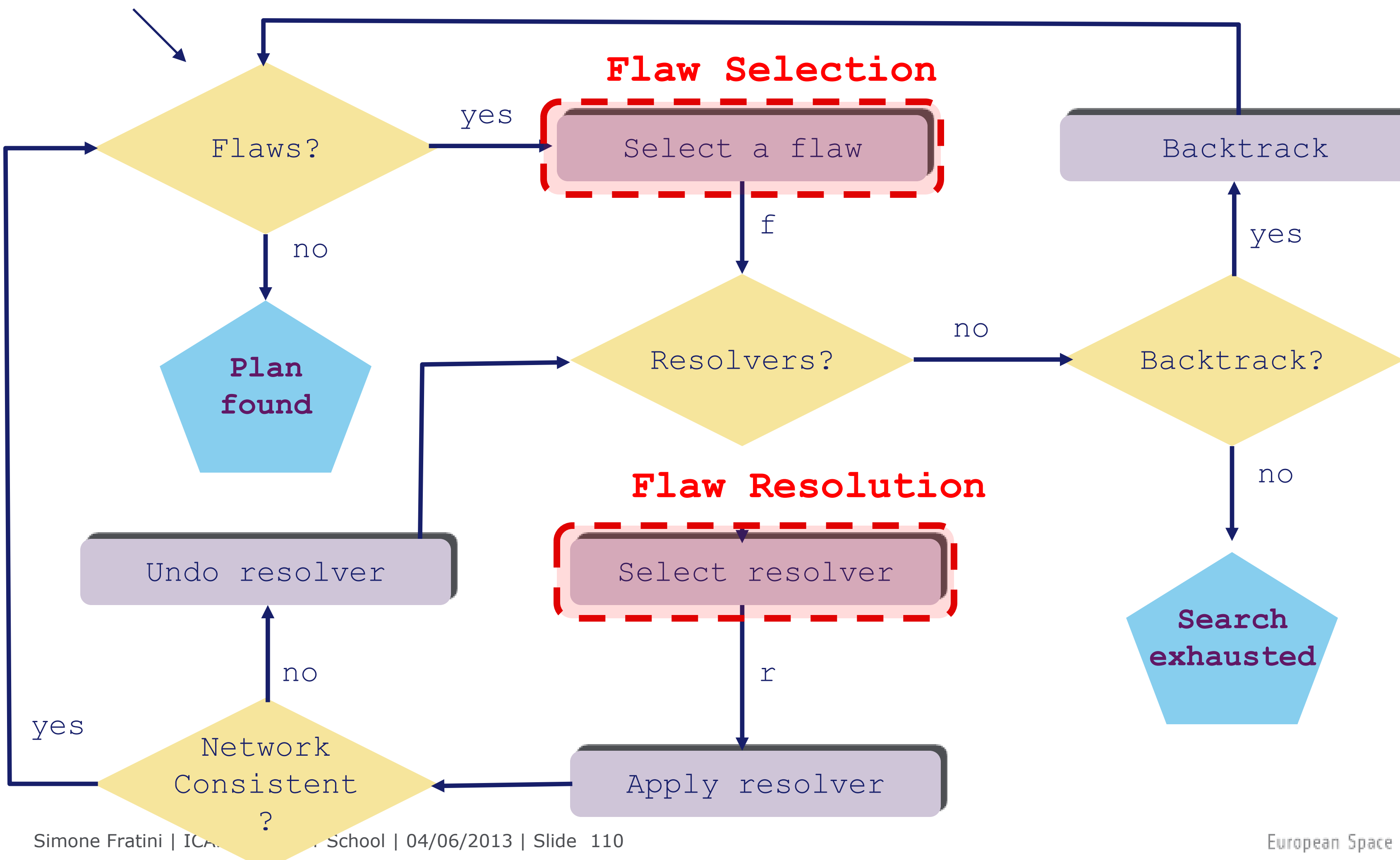
```
MicroscopicImager::TakeImage{
bool another_image;
if (another_image == true)
    meets(Place _pl);
   if (another_image == false)
    meets(Stow _st);}
```



Courtesy of S. Bernardini

Search algorithm

Depth-first search with chronological backtracking



ARCHITECTURES

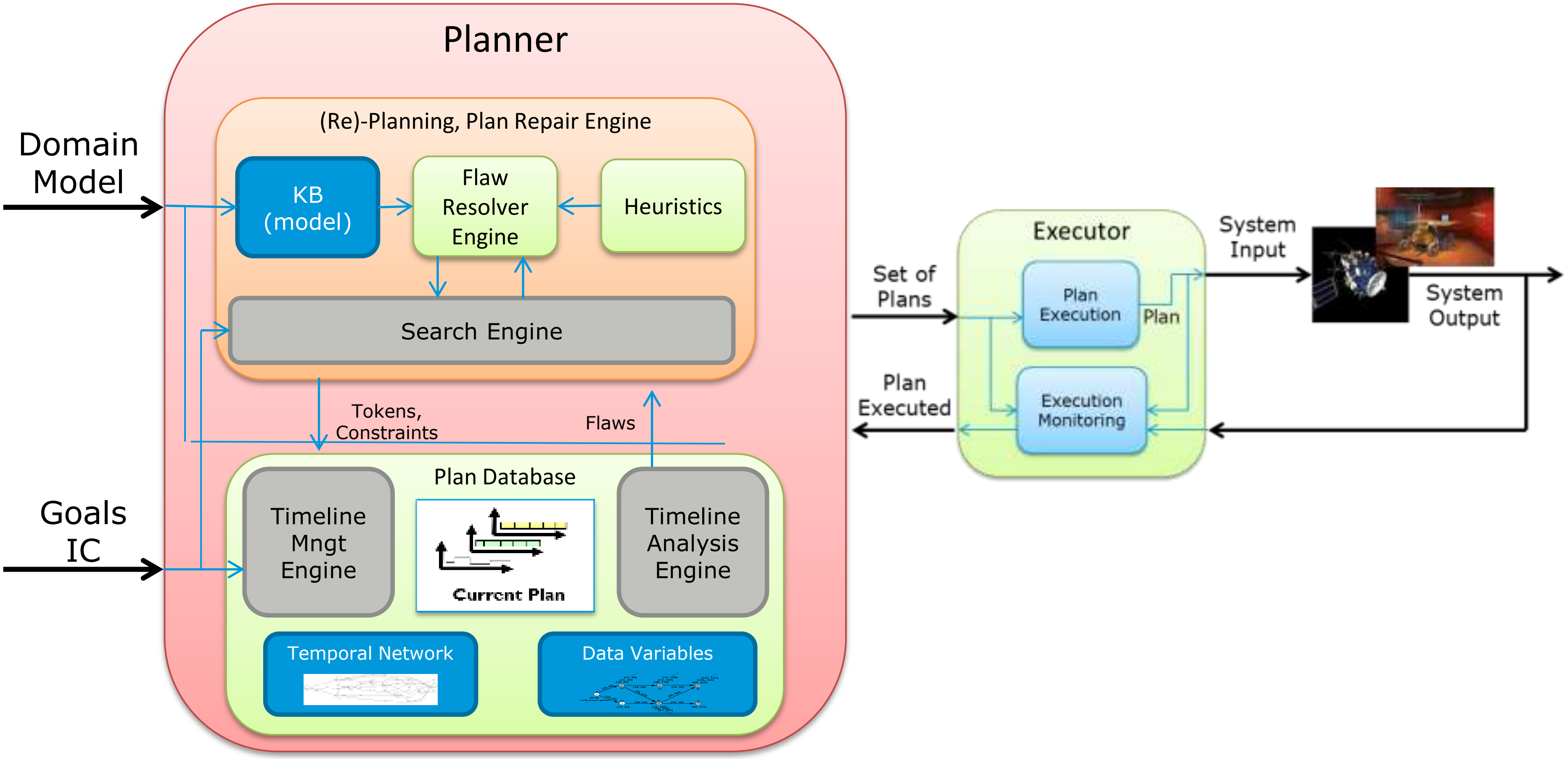
Section Summary

1. A generic architecture for timeline PSE (Planning, Scheduling and Execution)

2. Examples of real architectures (On Ground Segment)
   a. HSTS (since 1990) – NASA
   b. MAPGEN – NASA/JPL (2004)
   c. Mexar 2(2005), Raxem (2006) - ESA
   d. APSI Framework (2007), MrSpock, AIMS, XMAS - ESA

3. Examples of real architectures (On Board Segment)
   a. Remote Agent (1999) - NASA
   b. ASE, EO-1 (2003) – NASA/JPL
   c. GOAC – Goal Oriented Autonomous Controller (2011) - ESA

A Generic Architecture



HSTS

Hubble Space Telescope Scheduler

NASA - 1990

HSTS: Hubble Space Telescope Scheduler

1. Large set of targets and images
    a. Hundreds of proposals
    b. Thousands of targets
    c. Tens of thousands of images
2. Constraints on imaging activities:
    a. Visibility of targets
    b. Keep out areas (sun, moon, ground, etc.)
    c. Timing, setup constraints, etc.
3. Long term planning and plan updates
    a. Must permit analysis, changes, re-planning
4. Impractical and suboptimal to use humans

Thanks to Ari K. Jónsson, Reykjavík University

Technology - Spike (STSCI)

1. Hierarchical Request Scheduling
    a. Automated optimization search
        – Constraint-based scheduling + variety of heuristics
    b. Suitability Functions - constraints and preferences
    c. Plan windows – long-range and short-range

2. Capabilities provided
    a. Automated observation plan generation
    b. Toolkit for integration and visualization

3. Impact
    a. In use for HST scheduling since 1990
    b. Multiple missions have used Spike

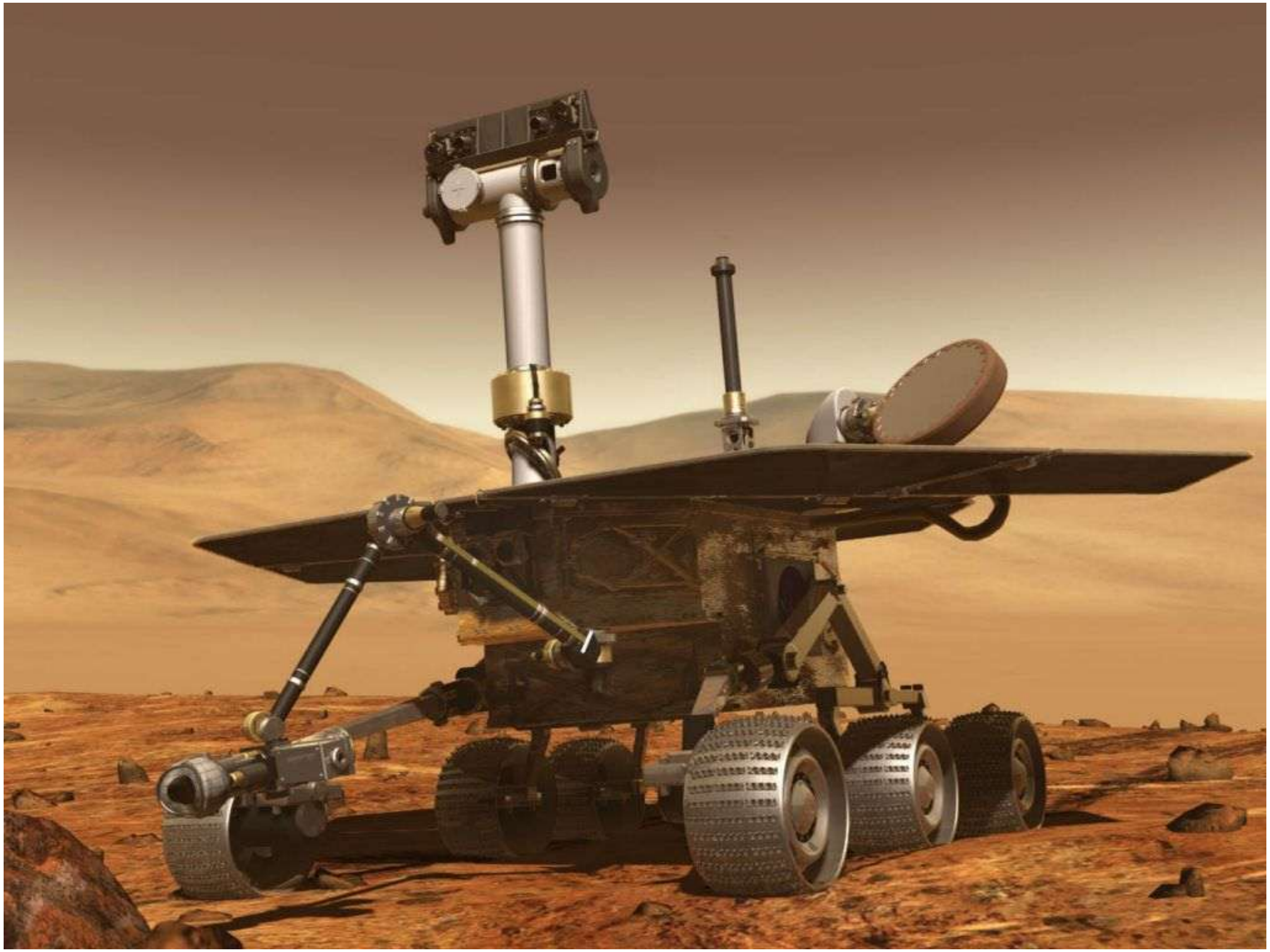
Thanks to Ari K. Jónsson, Reykjavík University

MAPGEN

Mixed Initiative Activity Planning Generator

NASA - 2003

Mars Exploration Rover (MER)



MER Operations

- Daily mission control cycle
  - Receive rover status and science data
  - Science and engineering analysis
  - Science team generates requests
  - Build plan from science and engineering requests
  - Approve activity plan
  - Turn plan into command sequences
  - Validate and approve sequence
  - Uplink to rover

- Time allowed
  - 18 hours for whole cycle
  - 1-2 hours for building and approving new plan

Thanks to Ari K. Jónsson, Reykjavík University

Technology – MAPGEN (NASA)

- Mixed-initiative Activity Plan Generator
  - MAPGEN: Existing JPL activity plan editor
  - EUROPA planning engine from Ames Res. Ctr.

- Capabilities provided
  - Active constraint rule enforcement
  - Levels of automated planning
  - Safe plan editing (moves, swaps, changes)
  - Automated activity generation

- Impact
  - Nominal mission: 15-40% increase in science data
  - Still in use today (but less critical)

Thanks to Ari K. Jónsson, Reykjavík University

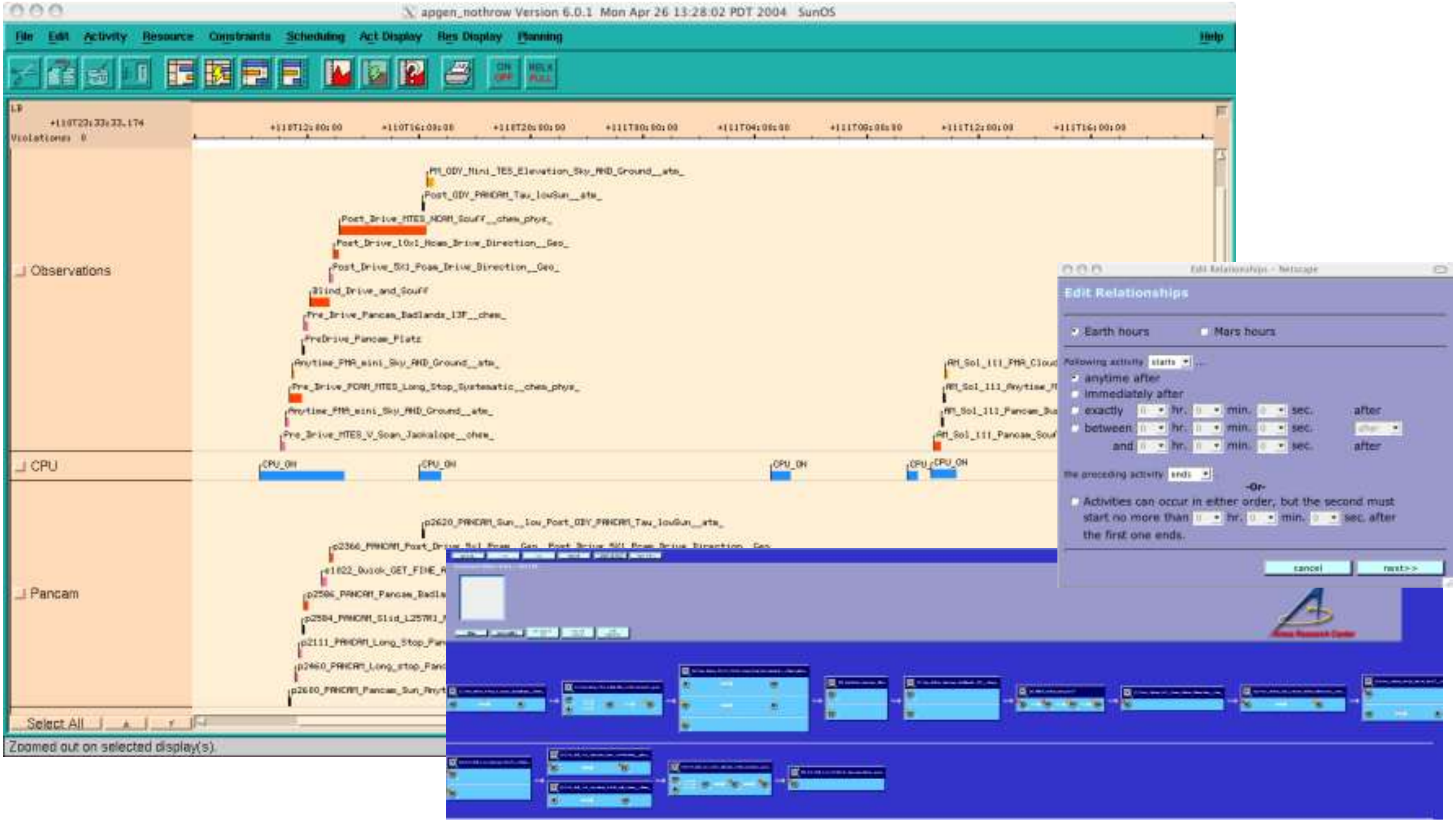
Underlying AI Technology

- EUROPA
  - General framework for planning and scheduling
  - Timelines-based plan representation
  - Complex constraints and resources
  - Dynamic constraint reasoning (incl. temporal)

- EUROPA in MAPGEN
  - Representation of plan and constraints
  - Active enforcement of plan constraints
  - Passive flagging of resource violations
  - Levels of automated planning and plan completion
  - Temporal reasoning used in Constraint Editor

Thanks to Ari K. Jónsson, Reykjavík University

Role of User Interface in MAPGEN



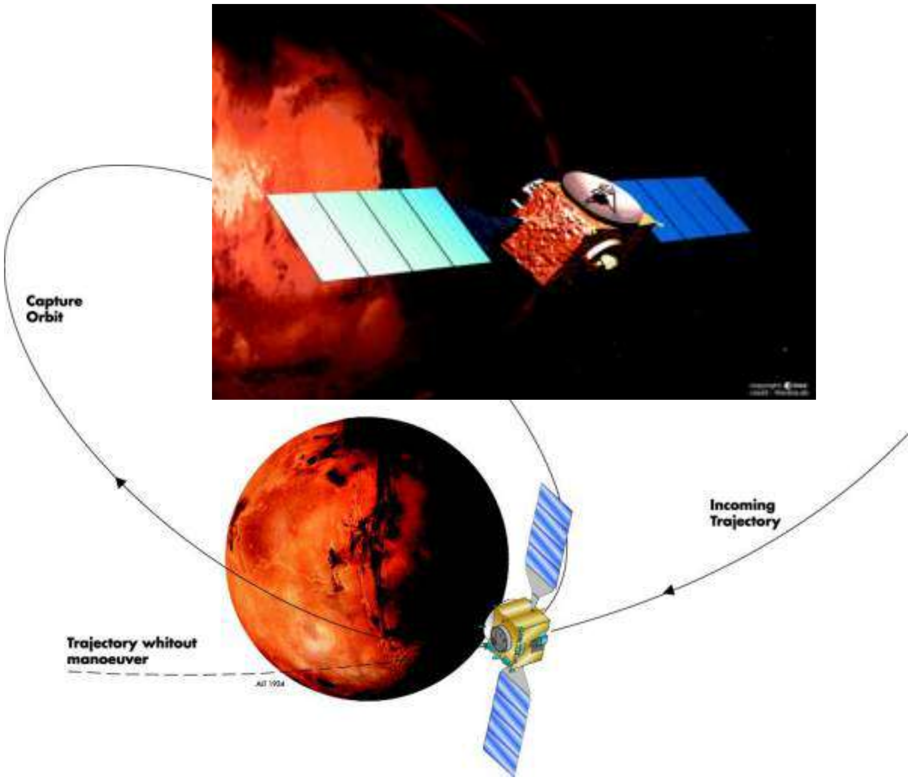
Mexar 2 - Raxem

Mars Express Memory Dumping and Command Uplink

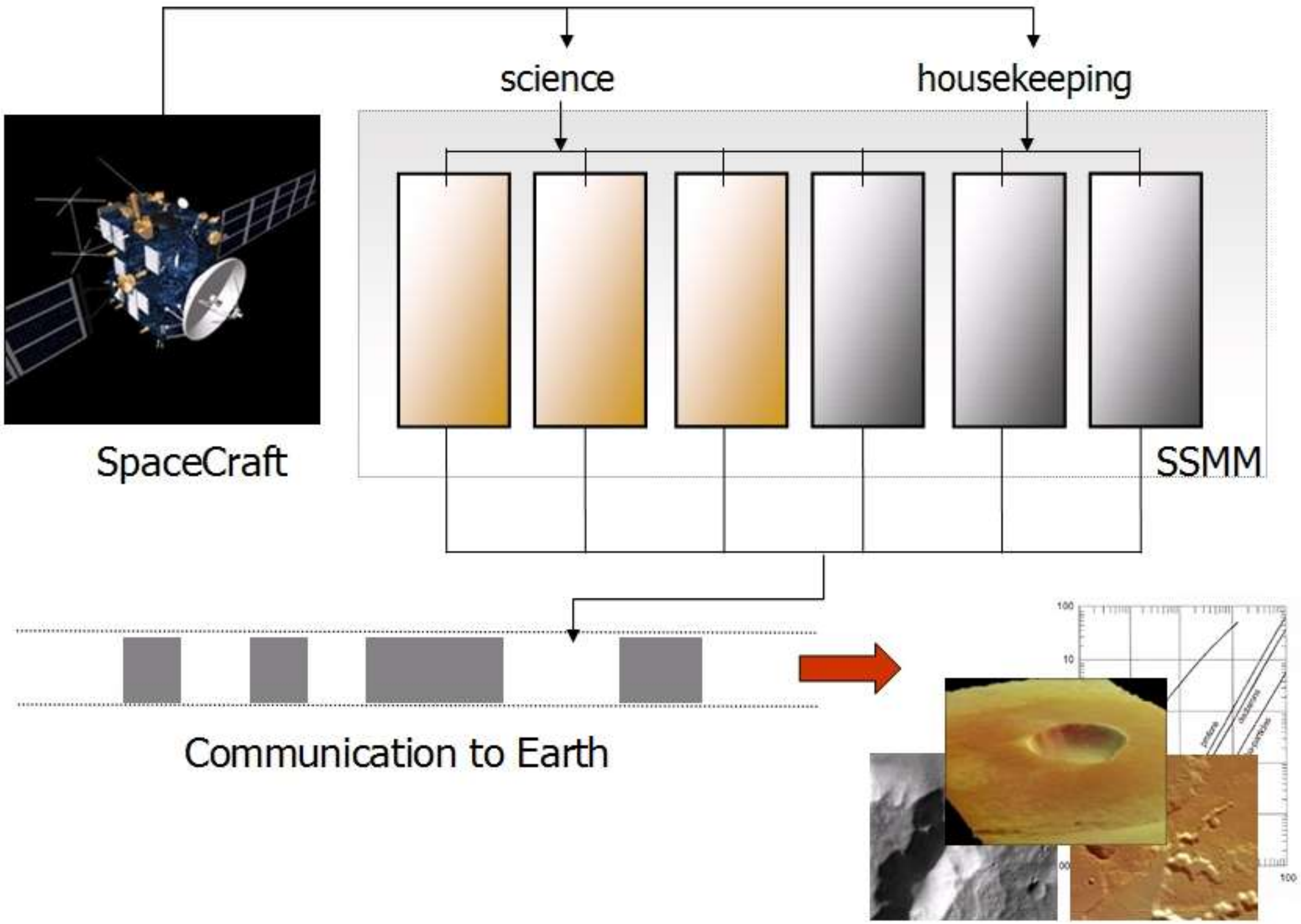
ESA – 2005, 2007

The Mars Express (MEX) mission

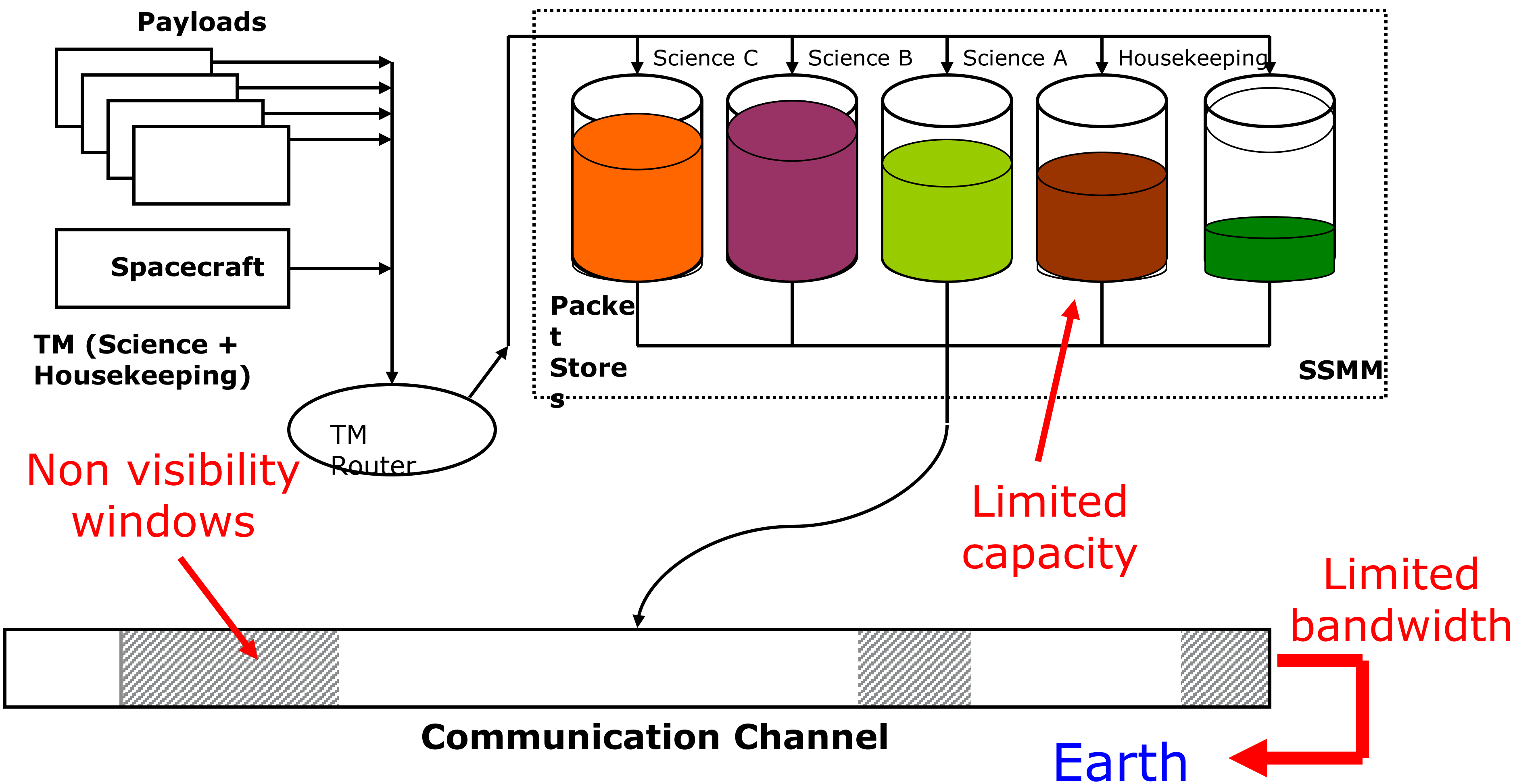
1. Launched on June 2003
2. The space probe is currently orbiting around Mars
3. Seven scientific payloads which collect data to study the Martian atmosphere and the planet's structure and geology
4. 2-3 Gb of data generated on daily basis



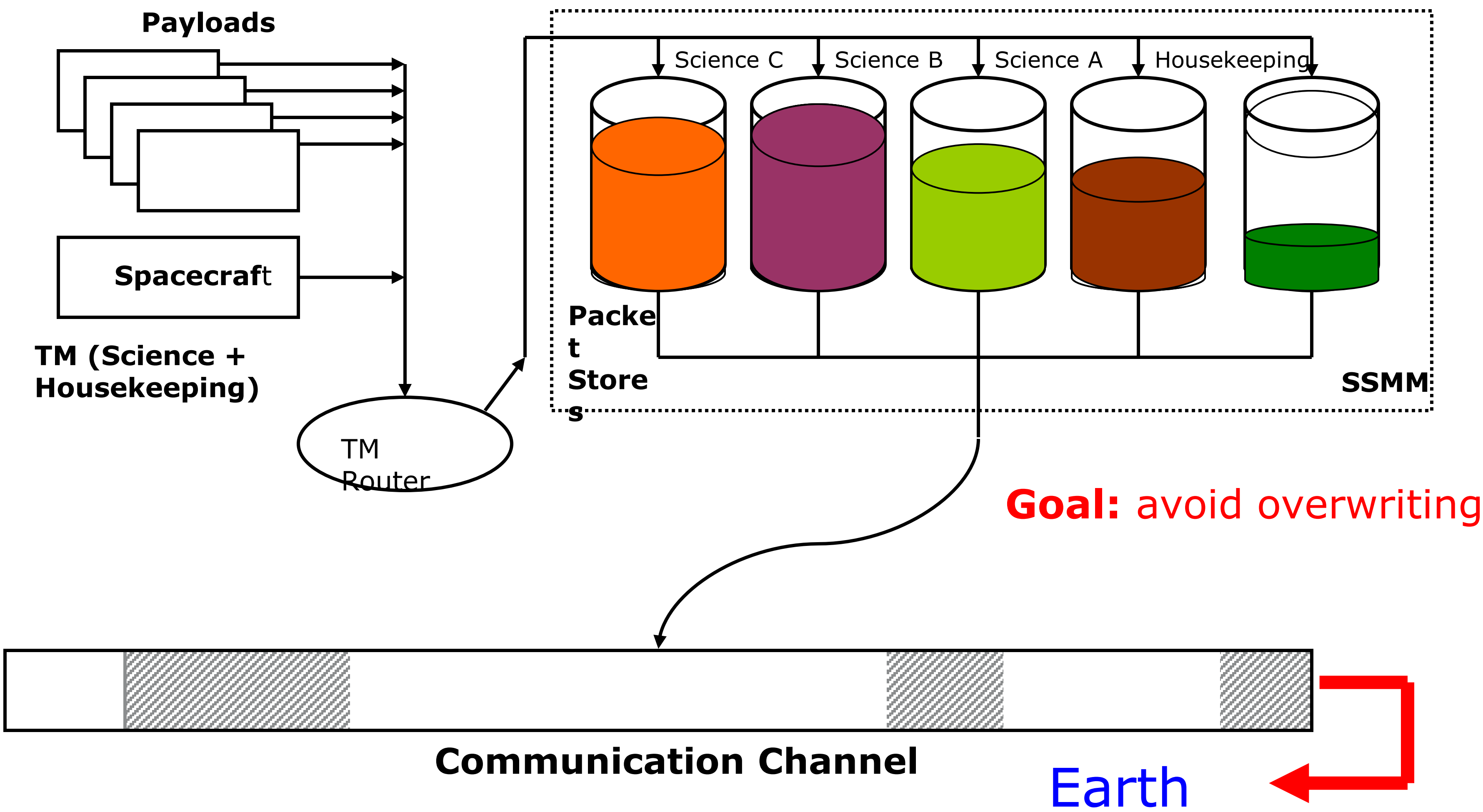
Relevant MEX sub systems



The MEX-Memory Dumping Problem



The MEX-Memory Dumping Problem



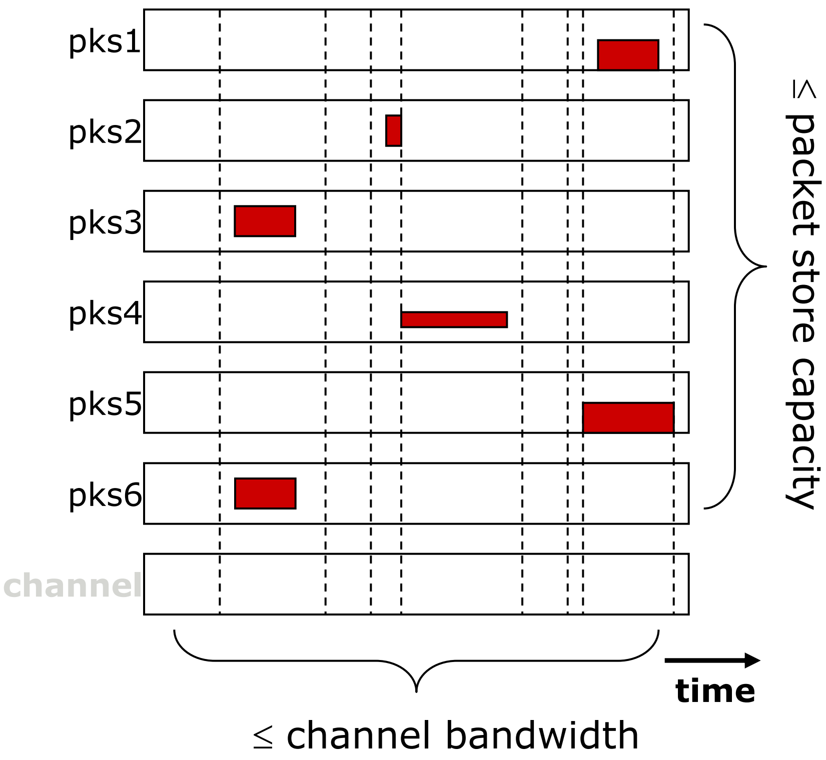
A Timeline-Based Model

1. For each packet store (and the communication channel) we define a different timeline

2. The temporal horizon is subdivided in contiguous time intervals such that instantaneous memory operations can happen only at the edges

3. Decision variables (flow values) represent the volume of data dumped within each interval

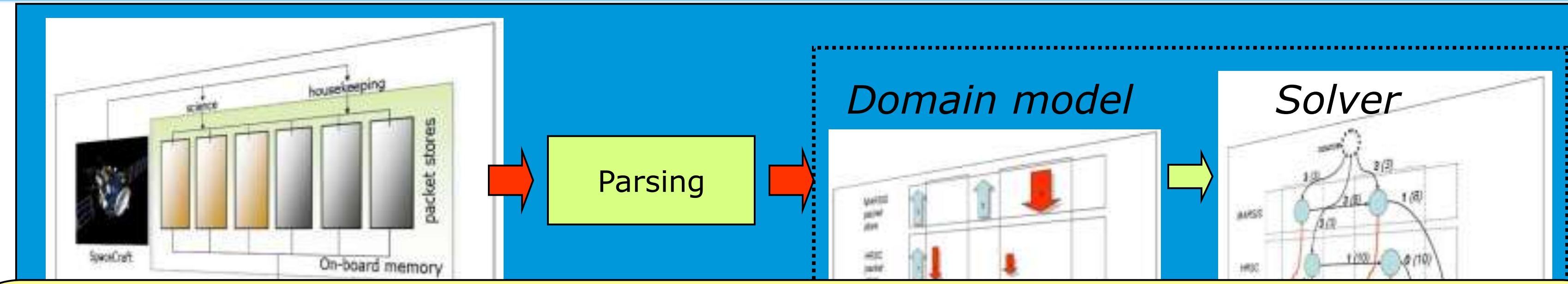
4. Two types of constraints:
   a. the ones imposed by the channel bandwidth and
   b. the ones by the packet stores capacity



Core Algorithm

1. Based on max-flow
    a. Complete algorithm: the dumping problem has a solution iff the maximal flow over the problem horizon is equal to total data stored
    b. Low polynomial cost: best known O(n2.5)
2. Housekeeping needs daily download
    a. Backtracking search to accommodate it before planning for science data
3. Additional Features:
    a. Guaranteeing continuity among adjacent planning intervals (management of residual memory store)
    b. Always generate a solution (relaxation)
    c. Improve configurability
    d. Allowing mission planners to specify single input events
    e. Control solution length
    f. Serve priority policy
4. For most of these features the "AI Model-Based" approach has helped tremendously

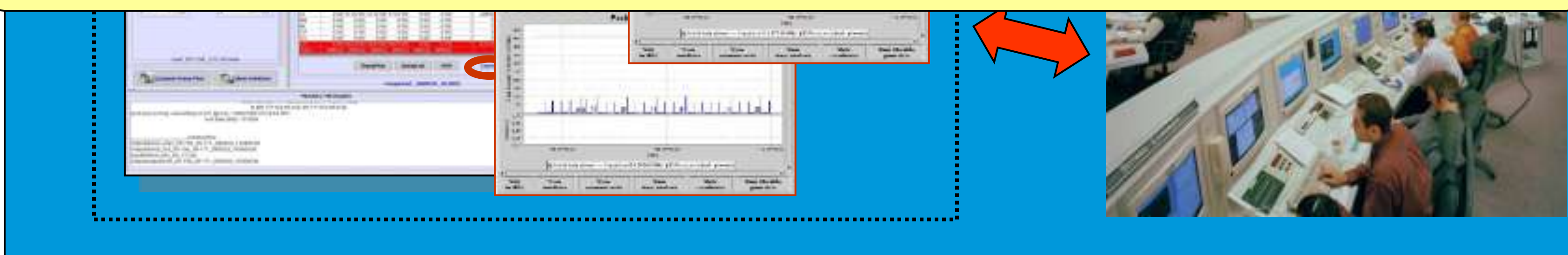
Final architecture: end-to-end cycle



Evaluation:

“The days before MEXAR2 are long gone and sometimes is hard to remember all the little details that made life miserable”

E. Rabenau (Mars-Express Planning Group)



APSI, MrSpock

Advanced Planning and Scheduling Initiative

MarsExpress Science Planning Opportunities Coordination Kit

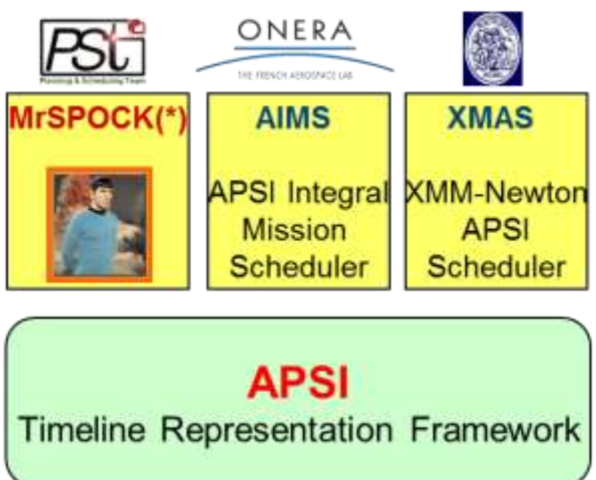
ESA – 2009

APSI & APSI-Related Activities

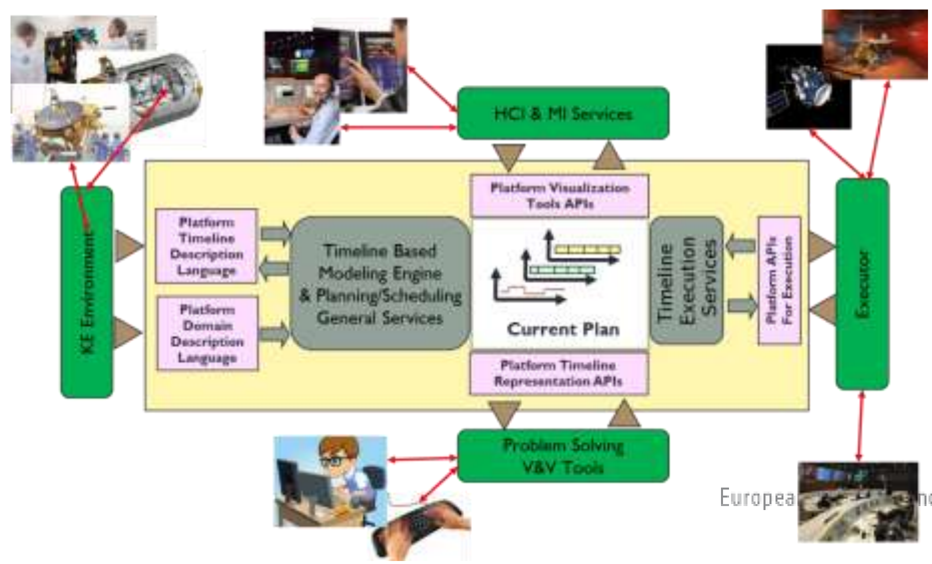
1. **Long Term Goal**: to introduce in ESA the **timeline-based approach** going from applications designed and deployed on purpose to model-driven, domain independent based P&S.

2. **Methodological approach**: build a stable layer for **rapid prototyping** and **robust design** of timeline based planning and scheduling experiments and deployed applications.

3. **APSI Project Goals**:
    a. to facilitate deployment and operational validation of an AI approach for planning & scheduling, designing a **plug-in architecture** and **re-usable functional modules**
    b. Bridging the gap between **advanced AI P&S** techniques and ESA's **mission planning environments**



(*) MrSPOCK: the MarsExpress Science Planning Opportunities Coordination Kit



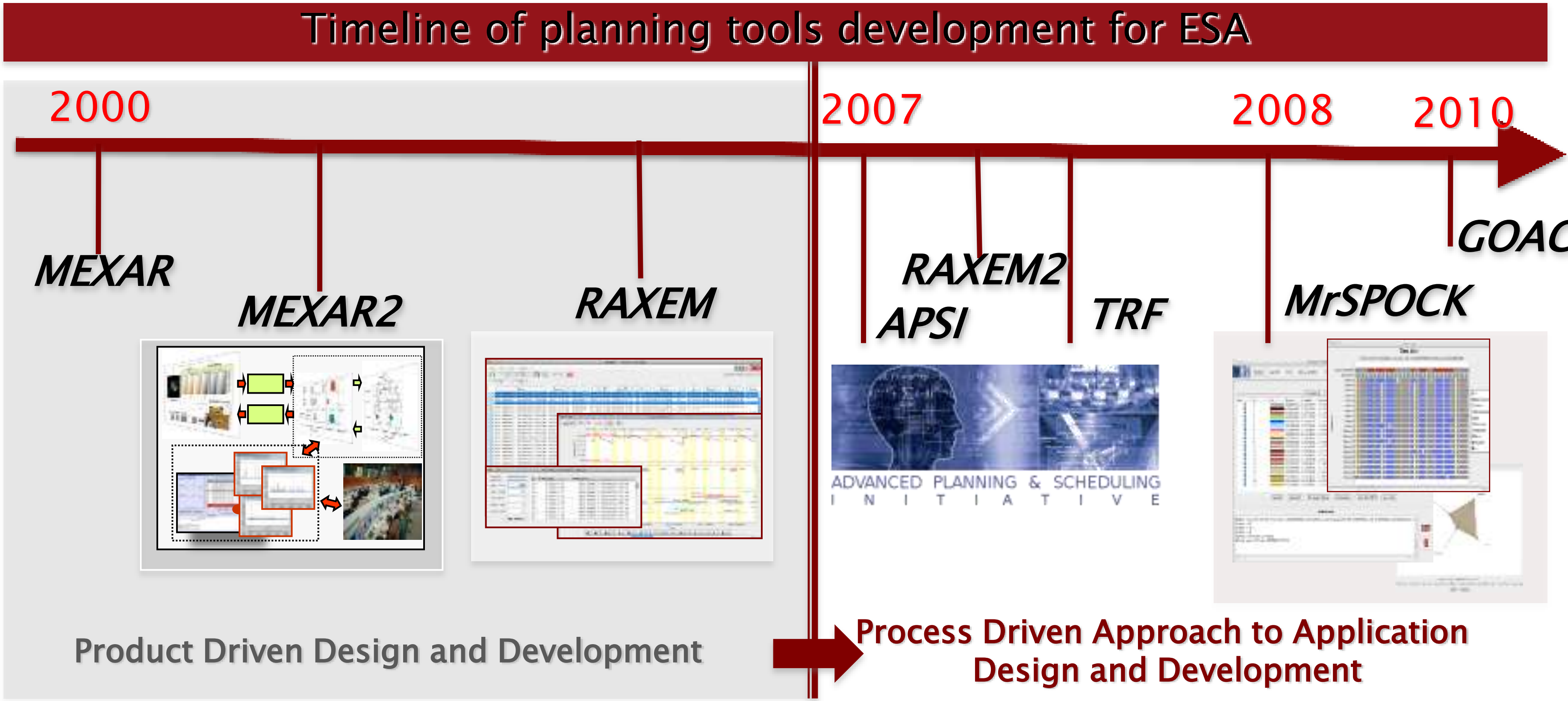
APSI & APSI-Related Activities

1. **Long Term Goal**: to introduce in ESA the **timeline-based approach** going from applications designed and deployed on purpose to model-driven, domain independent based P&S.

2. **Methodological approach**: build a stable layer for **rapid prototyping** and **robust design** of timeline based planning and scheduling experiments and deployed applications.

3. **APSI Project Goals**:
   a. to facilitate deployment and operational validation of an AI approach for planning & scheduling, designing a **plug-in architecture** and **re-usable functional modules**
   b. Bridging the gap between **advanced AI P&S** techniques and ESA's **mission planning environments**

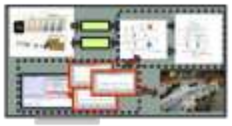
From Product to Process Driven



The ESA APSI Framework Timeline

2000

Mission-Specific P&S Applications
[2000-2007]



Mexar, Mexar2

Raxem

2007

[2007-2008]
Advanced P&S Initiative
APSI 1.0 Framework
Design & Deployment

APSI-Based P&S Applications
[2007-2009]

Mr SPOCK

AIMS

X-MAS

2009

APSI 2.0 Framework Enhancement
Support for Planning and Execution
Goal Oriented Autonomous Controller

GOAC
GOAL ORIENTED AUTONOMOUS CONTROLLER
[2009-2011]

DALA

3D-ROV

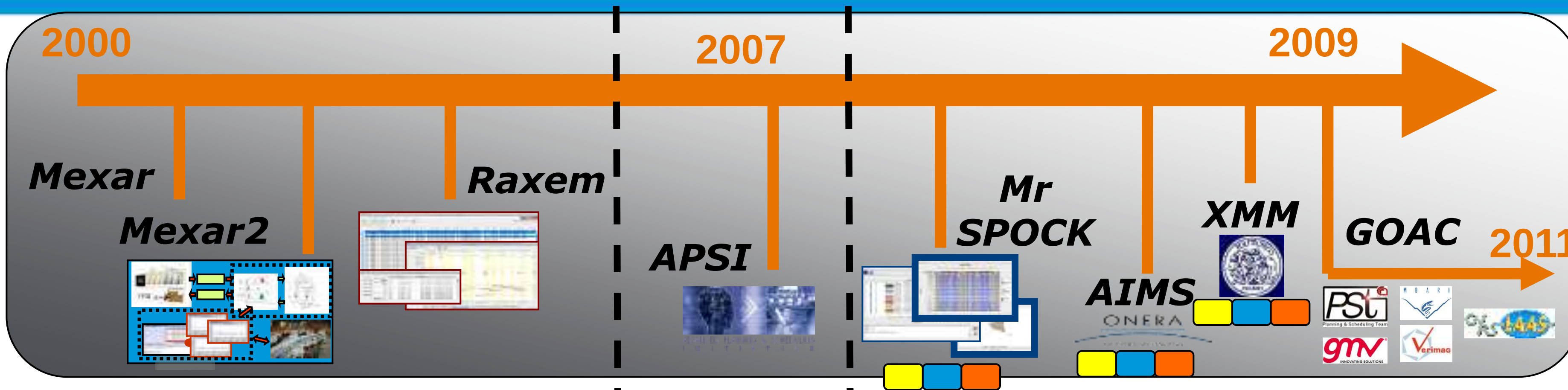
2011

IRONCAP
Innovative Rover Operations Concepts – Autonomous Planning
[2011-2012]

2012

APSI-KEE
APSI Knowledge Engineering Environment
Interactive Plan Execution & Control
[2012-2014]

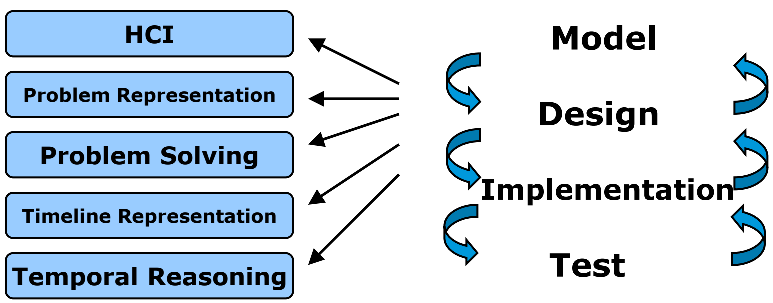
Why a Framework?

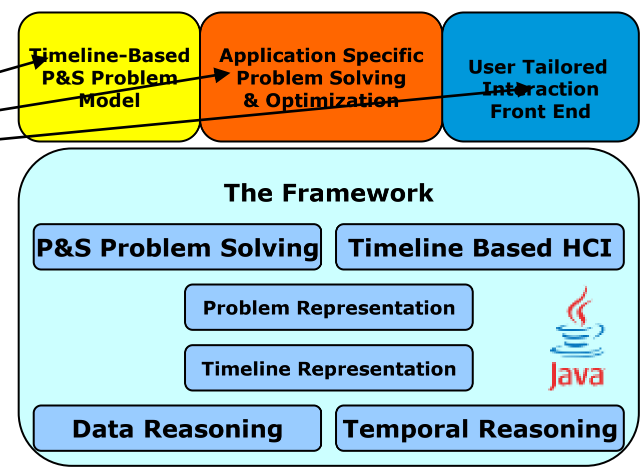


Applications built from scratch

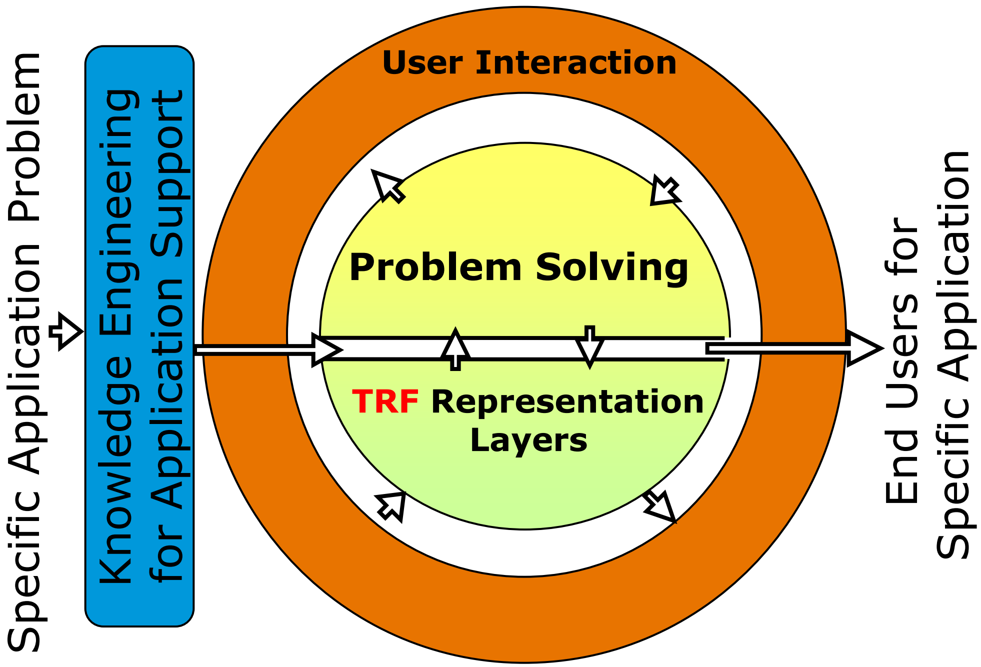
Framework Design & Implementation

Applications built On top of the Framework





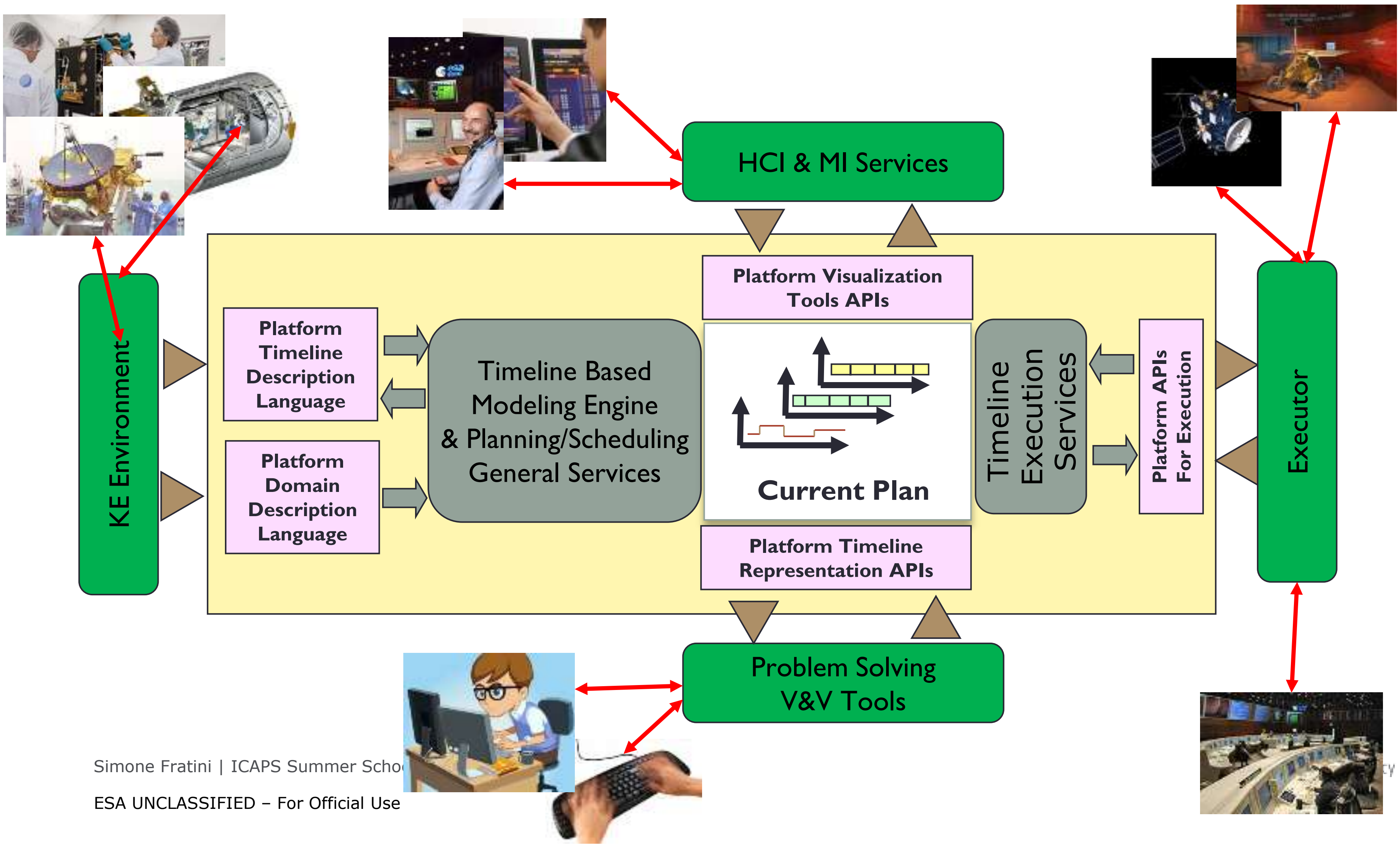
Timeline Representation Framework



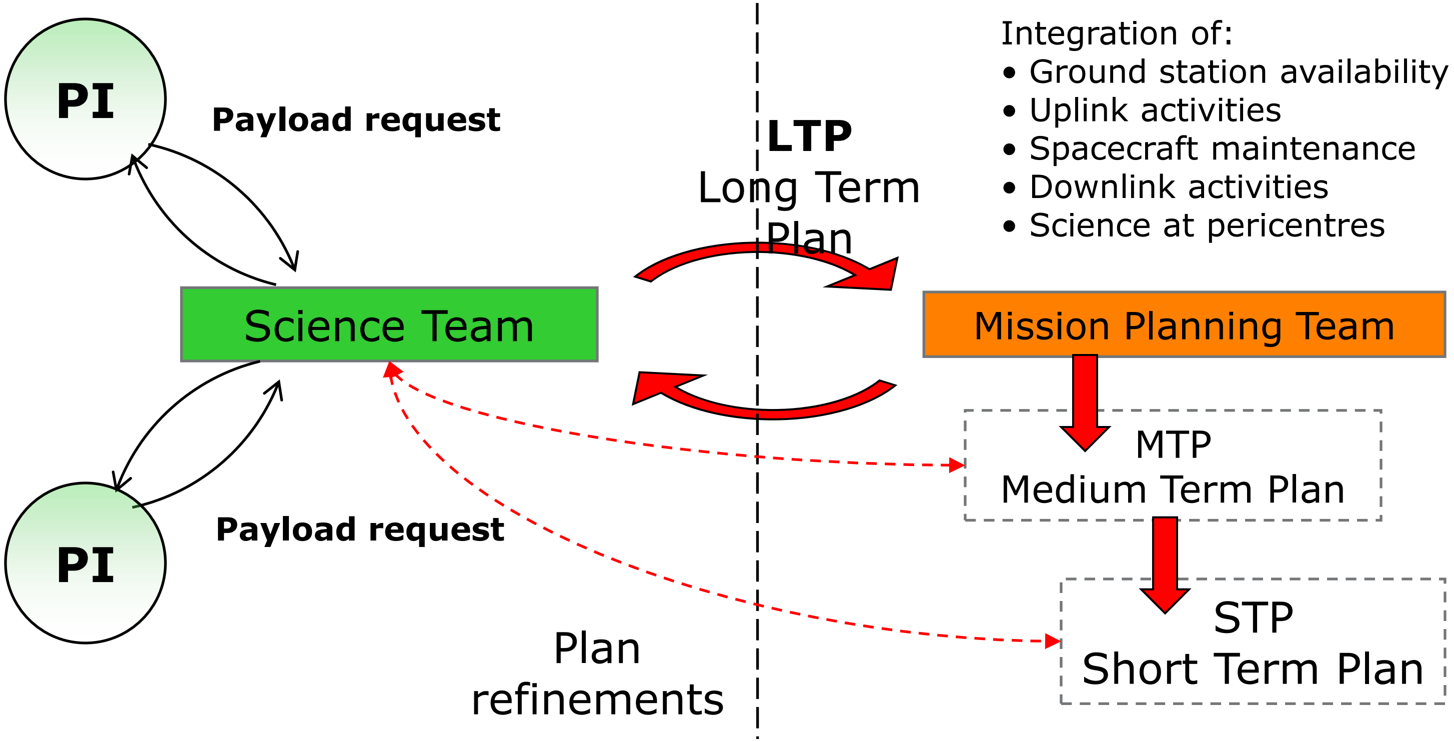
MrSPOCK	AIMS	XMAS
Mars Express Long Term Planning	APSI Integral Mission Scheduler	XMM-Newton Adv Mission Scheduler

APSI-TRF
Timeline Representation Framework

ESA-APSI Framework Architecture



Ma**r**sExpress **S**cience **P**lanning **O**pportunities **C**oordination **K**it



MEX-LTP: The Long Term Planning Problem

1. Generate a ***pre-optimized skeleton plan*** for Mars Express Long Term Planning

2. Aims:

   a. Minimize the iterations between Science Team and Mission Planning Team, taking into account a very detailed scenario and several co-existing constraints

   b. Provide the ability to explore the solution space according to different optimization functions
      - maximize planned science
      - maximize total UpLink/DownLink (UL/DL) time

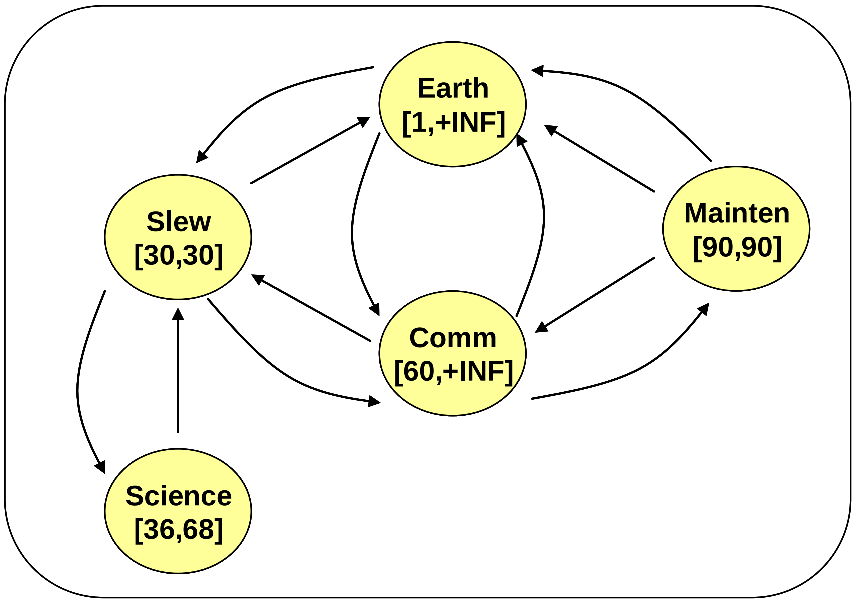
MEX-LTP: Critical points

1. Produce the target timeline whose slots (temporal intervals) contain:
    a. 1 4h uplink slot every 24h (alternative pattern : 2h/12h)
    b. 1 maintenance slot every 2 to 5 orbits (parametric)
    c. science and communication slots as much as possible

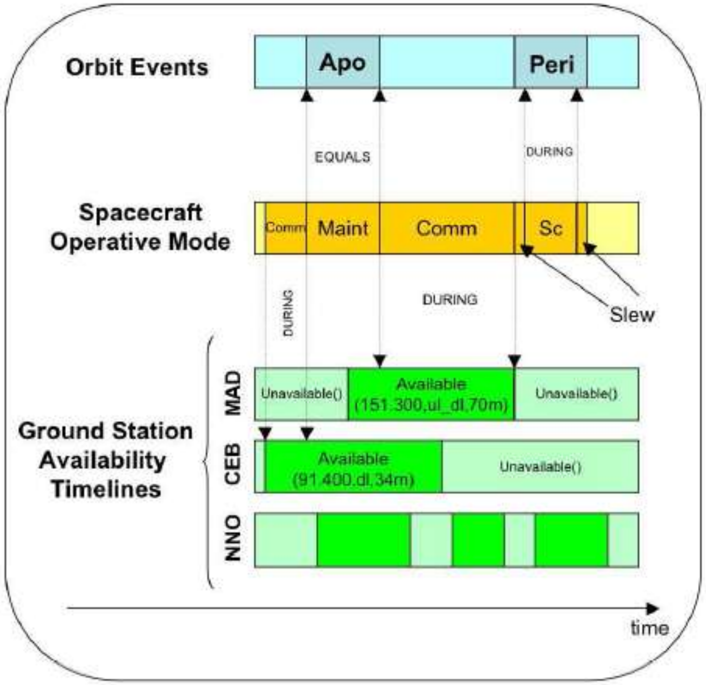
2. Problems for the solver side:
    a. Science requires Mars pointing, Communication requires Earth pointing
    b. Allocated Maintenance and Science must be synchronized with orbits pericenters and apocenters
    c. Communications must be scheduled against station visibilities
    d. Ground stations availabilities can overlap

3. Important problems for the users:
    a. Problem size is relevant (3 months of temporal horizon required)
    b. Mission is in its final phase so the following should be taken into account:
        - current practice is consolidated in terms of shape of solutions ==> "I would like to see the solution type I'm used to"
        - but, at the same time, users would like to just "push a button and have it all done"

Timeline-based Problem Modeling

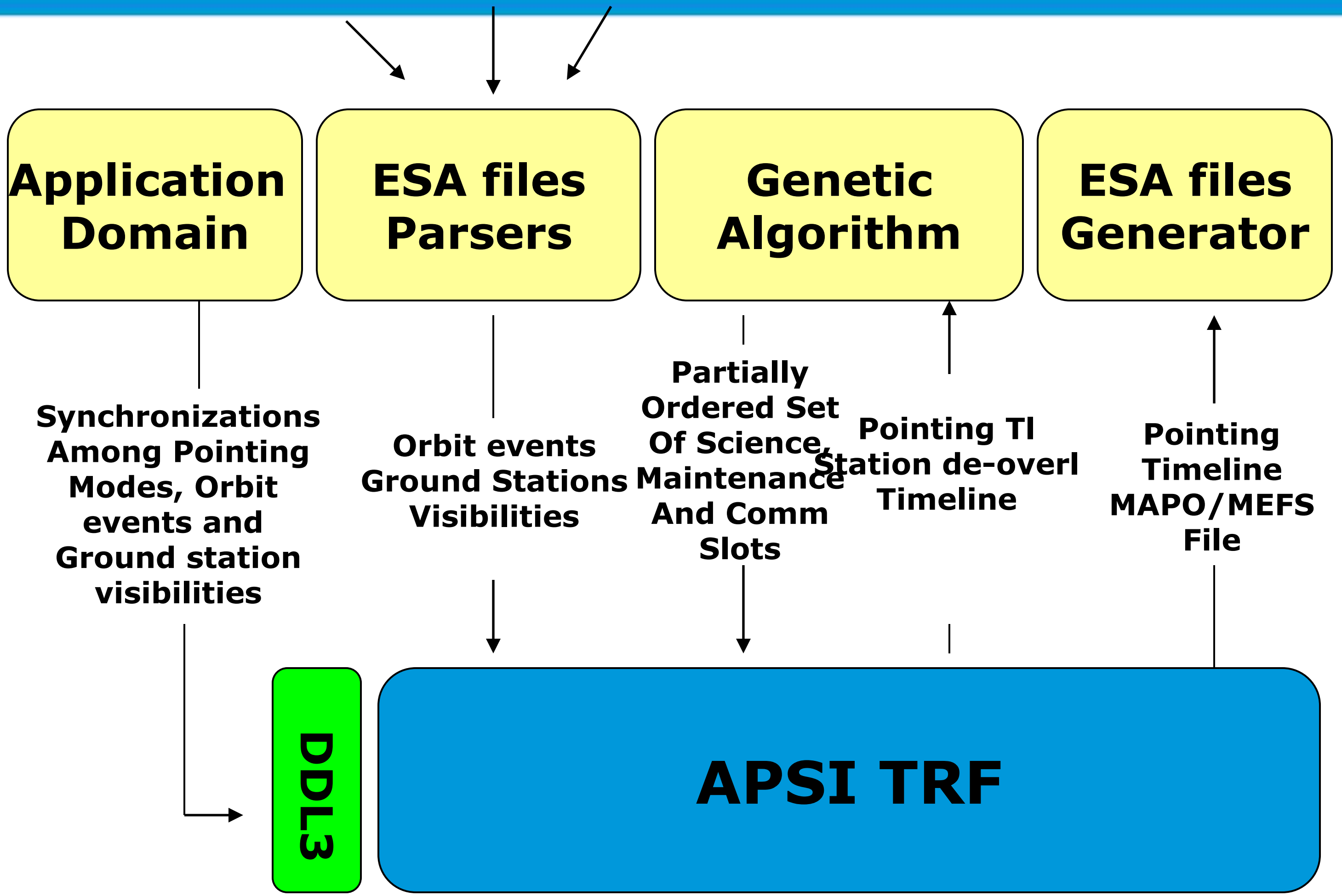


Correct mode transitions on the Timeline to be planned

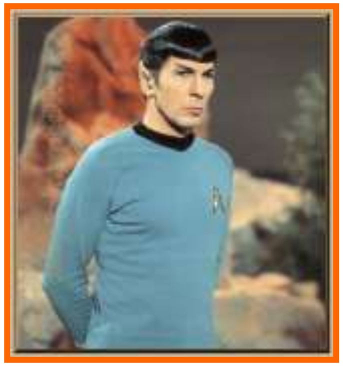


Synchronizations with pre-defined
Mission information

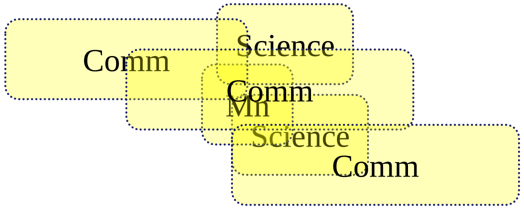
APSI-TRF Based Approach

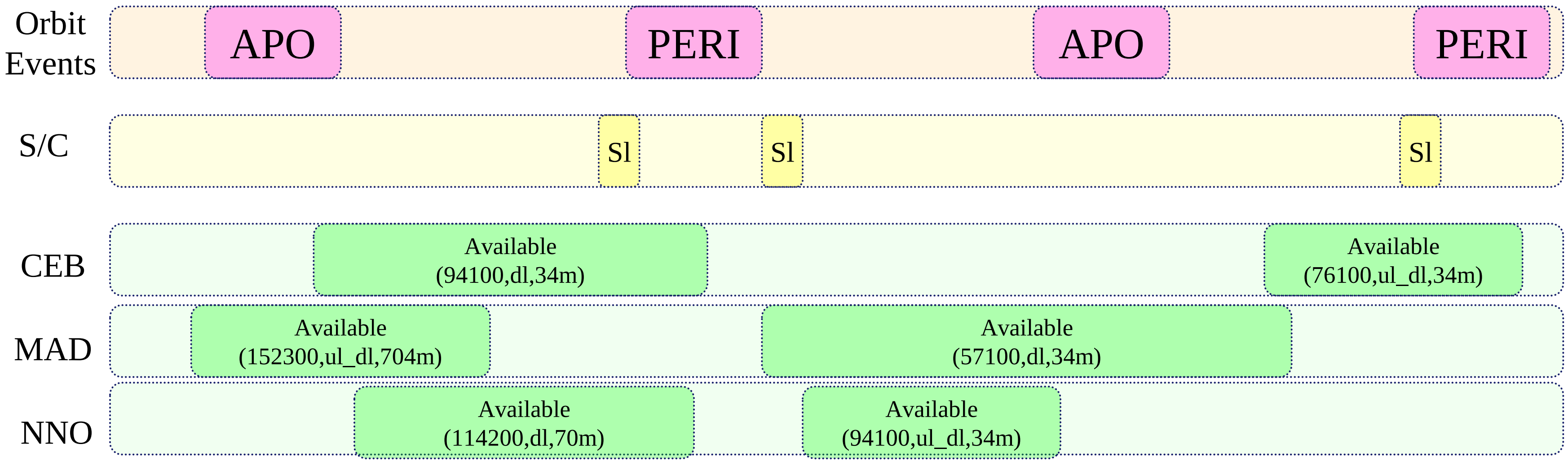


Problem Solving

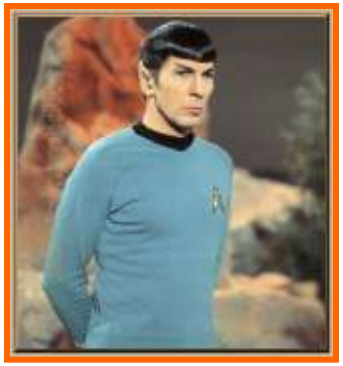


- Flight Dynamics
- Station visibility windows
- Planning Science/Mn Allocation
- Choosing Comm Wn and Station Deoverlaping
- Plan Completion
- Solution Extraction

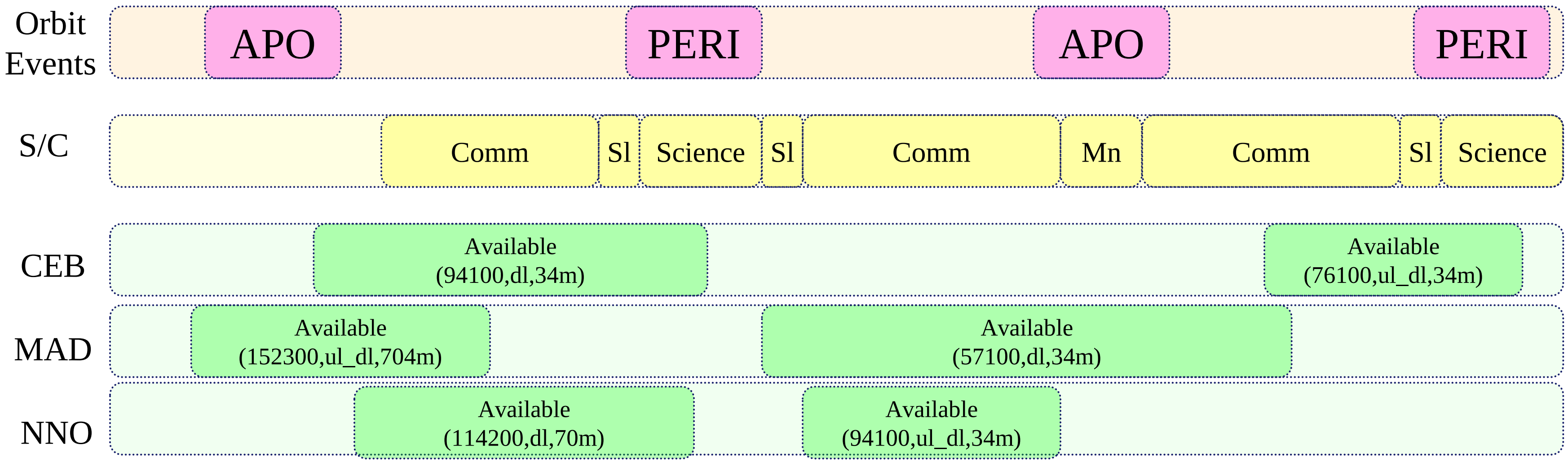




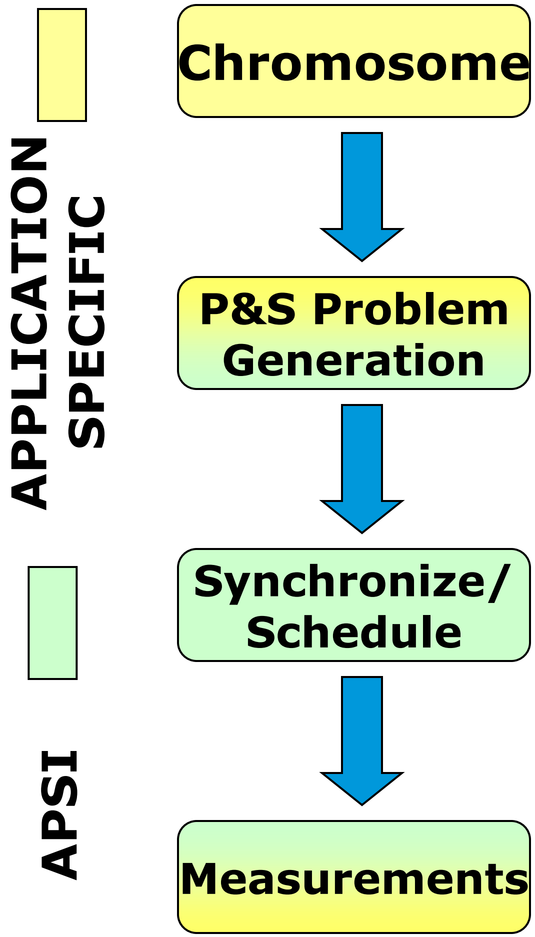
Problem Solving



- Flight Dinamic
- Station visibility windows
- Planning Science/Mn Allocation
- Choosing Comm Wn and Station Deoverlaping
- Plan Completion
- Solution Extraction



Solving Algorithm [Cesta&al08]



A genetic optimization is performed over an initial set of chromosomes which represents a pattern of science and maintenance allocation slots. This pattern fulfills the requirement of 1 maintenance slot every 2/5 orbits.

We build a network of time-flexible P&S problem containing
- 1 4h uplink slot every 24h (or 2h/12h)
- Science and Maintenance slots (driven by chromosomes)

Allocated Maintenance and Science are synchronized with pericenters and apocenters, communications are scheduled over station visibilities using the domain independent architecture's capabilities.

Measures (for fitness evaluation):
- Max # pericentres available for science
- Max total UL/DL time

Remarks on MrSpock

1. The choice of focusing the interaction on the concept of **timelines** which evolve over time has been successful

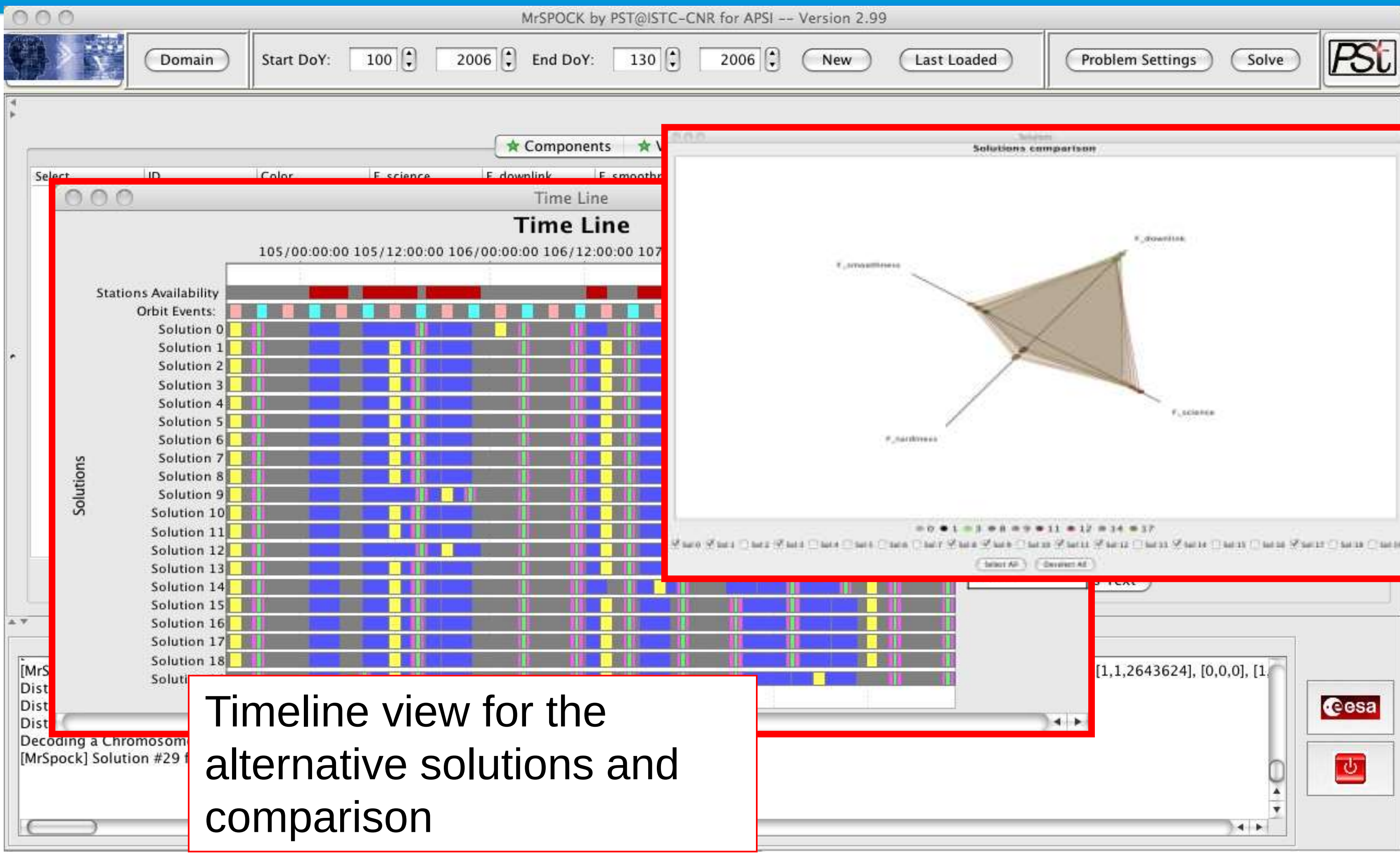
2. Temporal representation for the **timeline** is **very close to the way of taking decision in space contexts**

3. **Users' involvement** in problem solving has been fostered:
    a. users wants to be relieved by hard work,
    b. users also want solutions according to their preferences and expectations

4. Alternative solutions are greatly appreciated, as well as the instruments to **evaluate** them

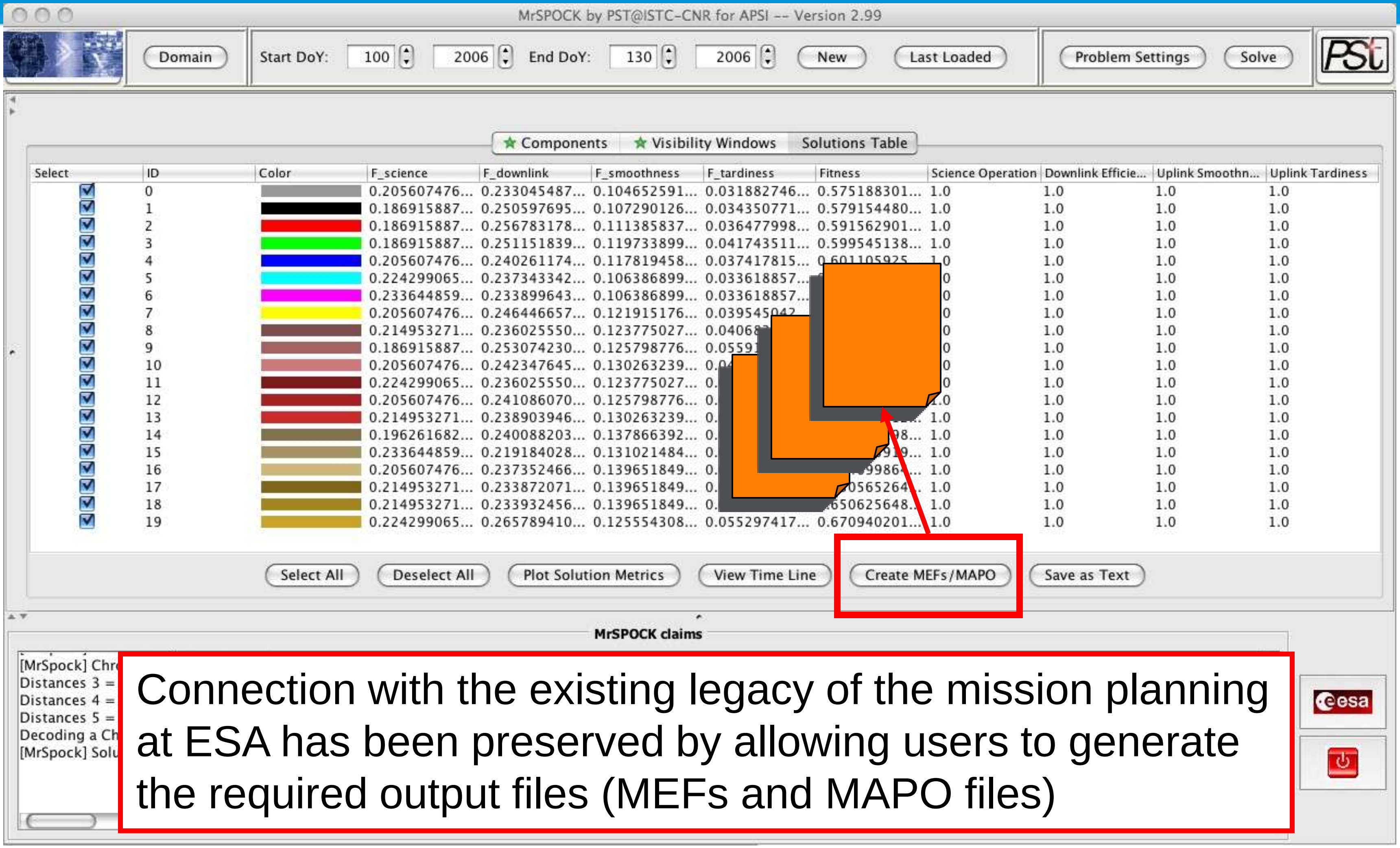
5. Need for tools/functionalities that enable easy modeling of problems (underlying **knowledge representation problem**)

User Interaction in MrSPOCK



Timeline view for the alternative solutions and comparison

User Interaction in Mr SPOCK



Connection with the existing legacy of the mission planning at ESA has been preserved by allowing users to generate the required output files (MEFs and MAPO files)

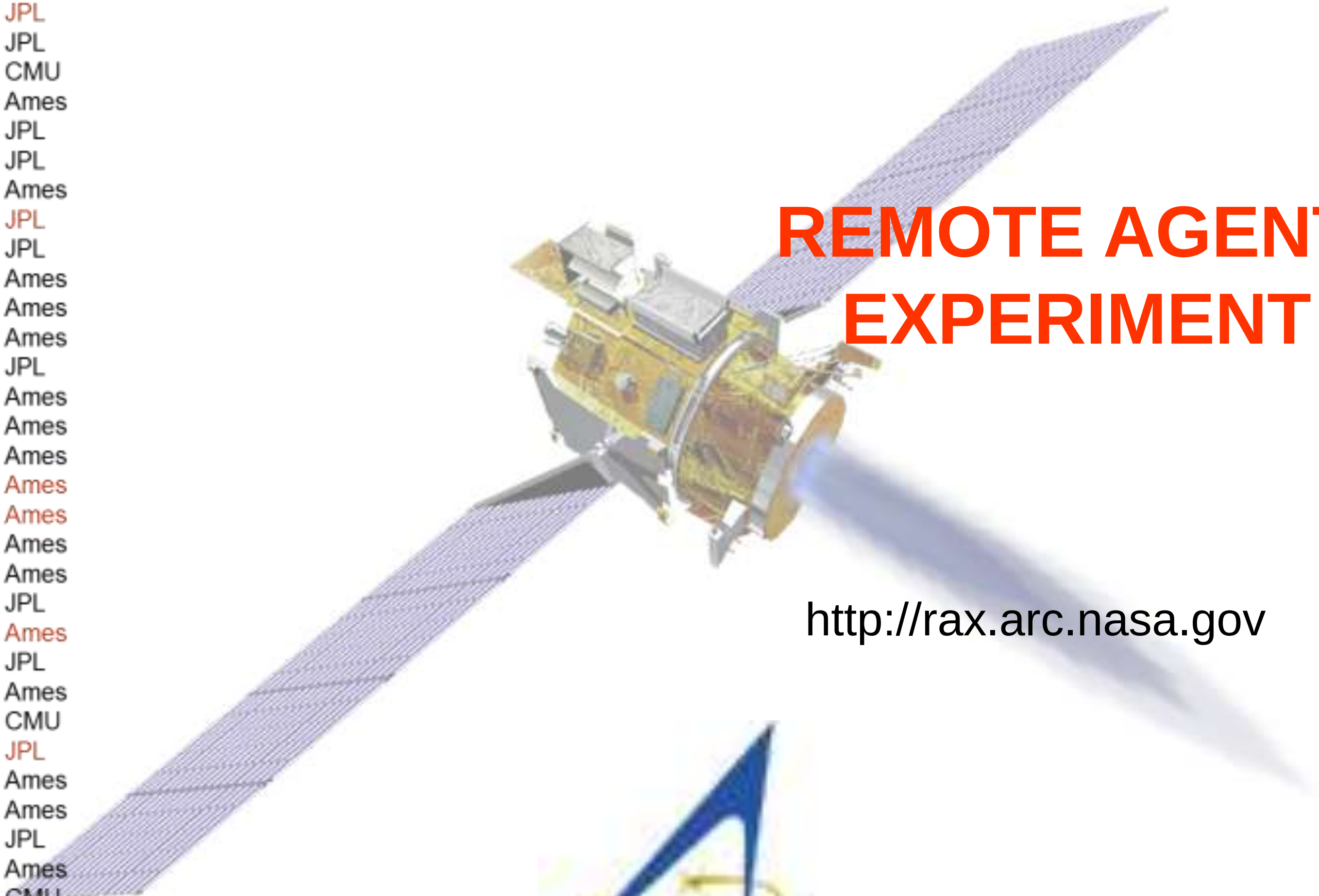
RAX-PS

Remote Agent Experiment

NASA – 1999

Remote Agent Experiment

Douglas E. Bernard	JPL
Steve A. Chien	JPL
Scott Davies	CMU
Gregory A. Dorais	Ames
Richard Doyle	JPL
Dan Dvorak	JPL
Charles Fry	Ames
Edward B. Gamble Jr.	JPL
Erann Gat	JPL
Bob Kanefsky	Ames
Ron Keesing	Ames
Jim Kurien	Ames
Guy K. Man	JPL
William Millar	Ames
Sunil Mohan	Ames
Paul Morris	Ames
Nicola Muscettola	Ames
P. Pandurang Nayak	Ames
Barney Pell	Ames
Christian Plaunt	Ames
Greg Rabideau	JPL
Kanna Rajan	Ames
Nicholas Rouquette	JPL
Scott Sawyer	Ames
Reid Simmons	CMU
Benjamin Smith	JPL
Gregg Swietek	Ames
William Taylor	Ames
Yu-Wen Tung	JPL
Michael Wagner	Ames
Greg Whelan	CMU
Brian C. Williams	Ames
David Yan	JPL



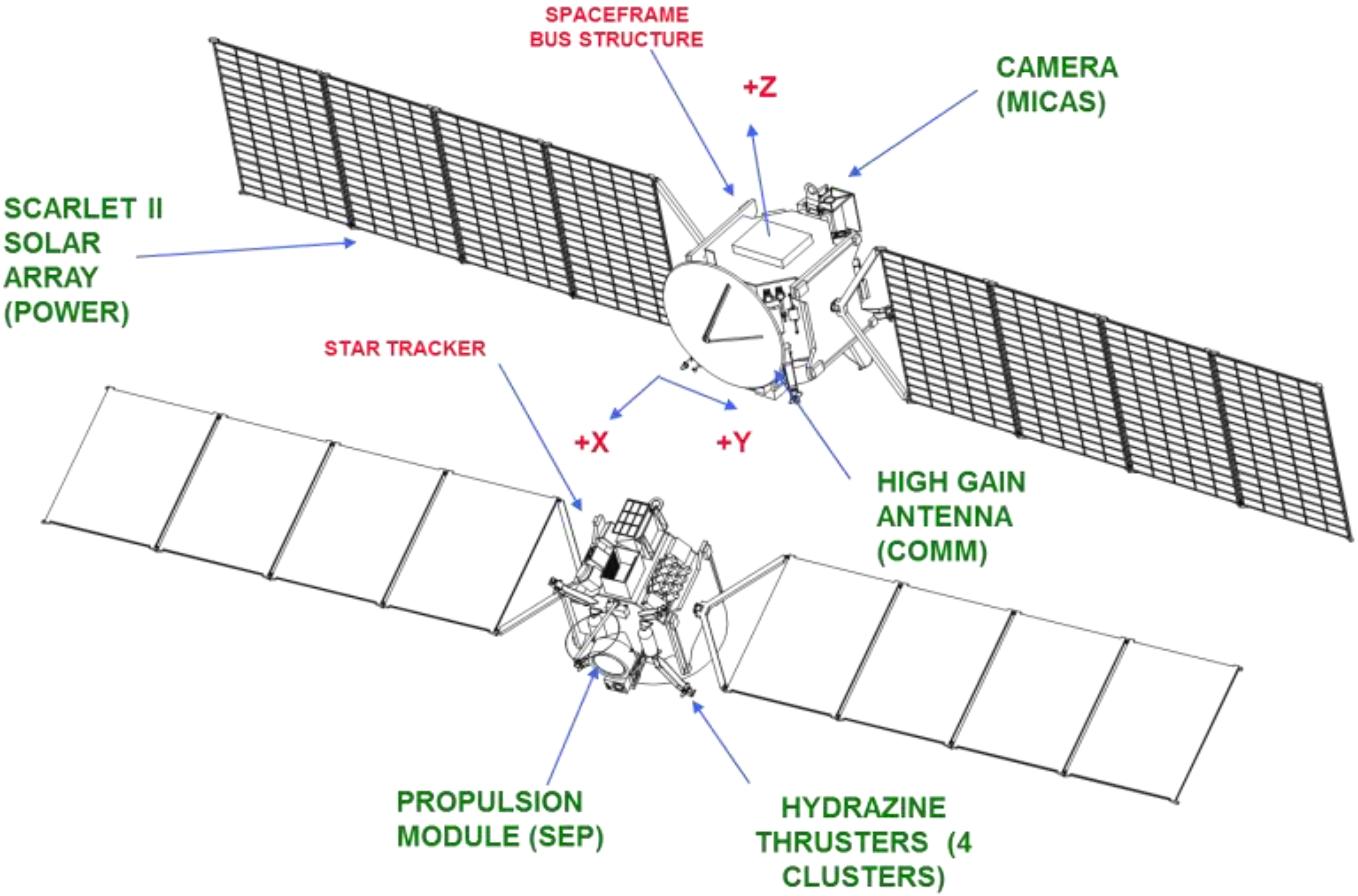
REMOTE AGENT EXPERIMENT

http://rax.arc.nasa.gov

Ames Research Center

K. Rajan – Presentation @ ESA – Paris, Sept. 2000

Deep Space 1 Spacecraft



K. Rajan – Presentation @ ESA – Paris, Sept. 2000

Deep Space 1 Mission

1. Remote Agent Experiment
   May 17-21, 1999

2. DS1: First New Millennium mission

3. 12 new technologies
   - Ion Propulsion System (IPS)
   - On-board optical navigation
   - Beacon experiment, ...

4. Fly-by of asteroid (July 01)

5. Remote Agent on DS1 wins NASA's 1999 Software of the Year

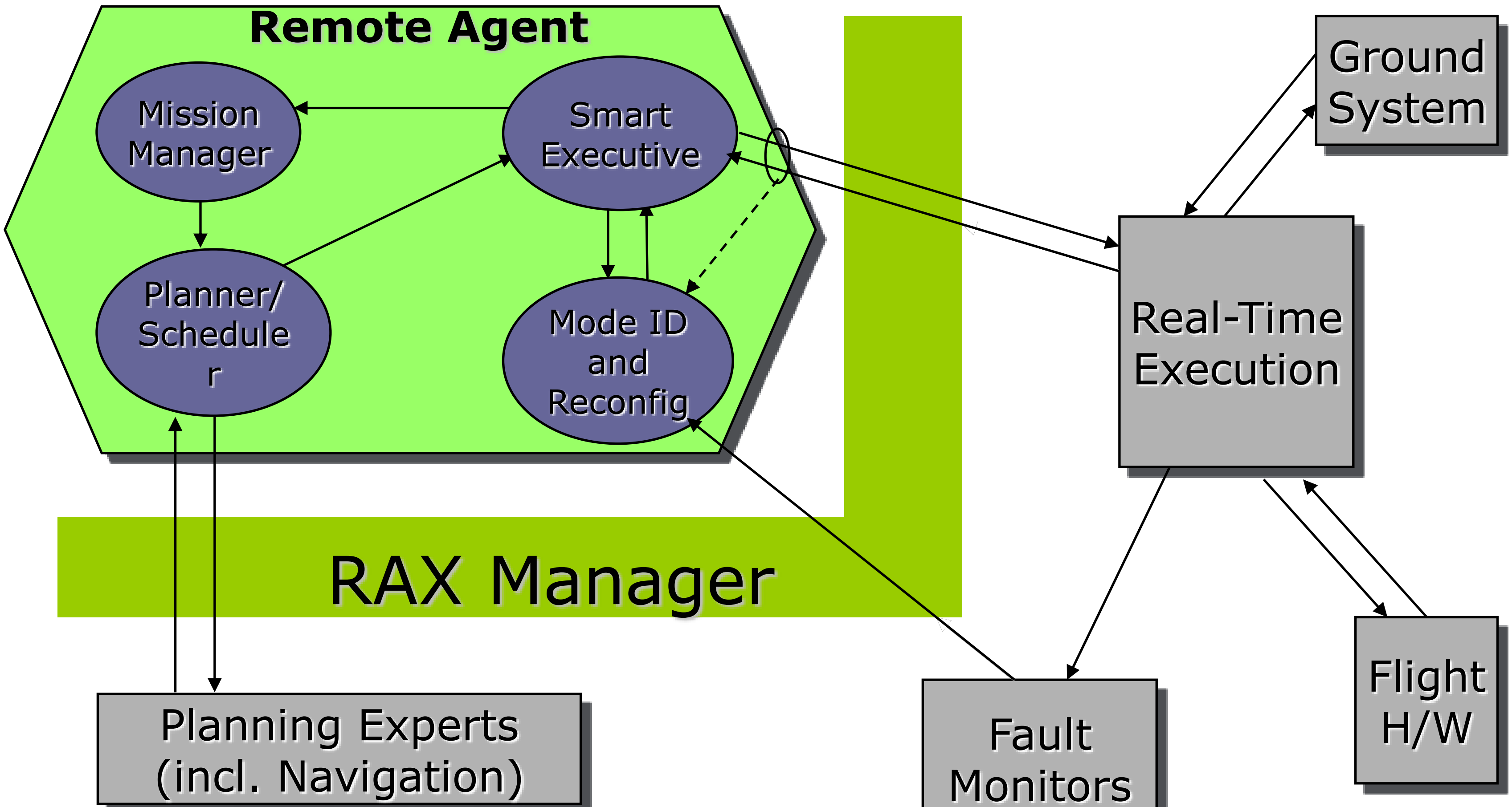
The RAX Experiment

1. The Remote Agent RAX controller, developed at NASA-AMES research lab took control of the space-ship DeepSpace-1 as a test of AI's ability to provide autonomous control.

2. RAX combines:
    a. space-system monitoring and sensing
    b. automatic, onboard planning,
    c. execution monitoring,
    d. fault diagnosis, and
    e. re-planning

   in an integrated system.

Remote Agent Experiment: Architecture



K. Rajan – Presentation @ ESA – Paris, Sept. 2000

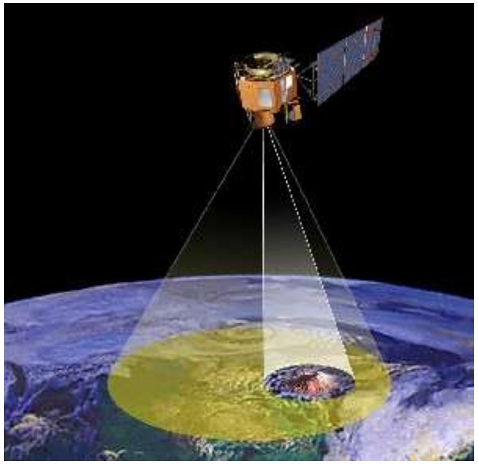
ASE / EO-1

Autonomous Science Experiment

Earth Obersving-1

NASA/JPL – 2003

Autonomous Science Experiment (ASE)

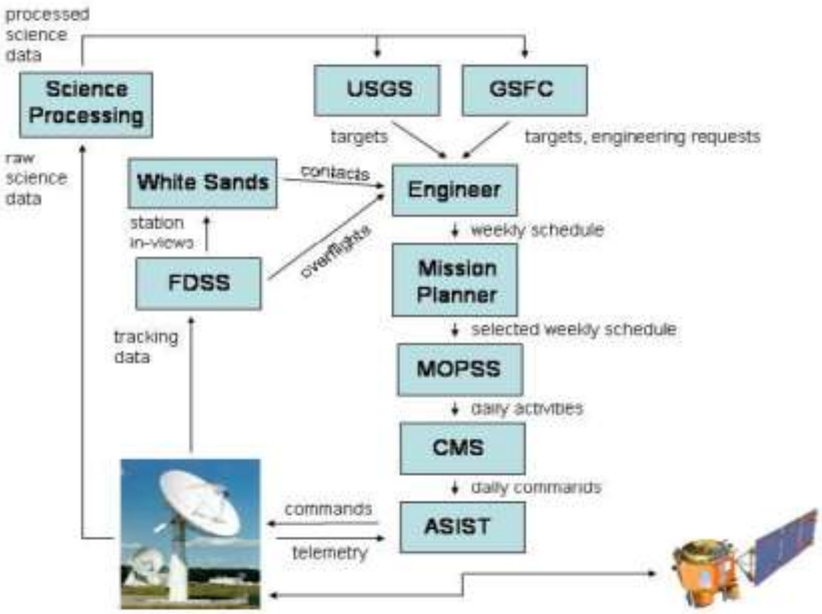


1. Scope:
   a. Control of primary activities
   b. Management of schedule
   c. On-board data analysis

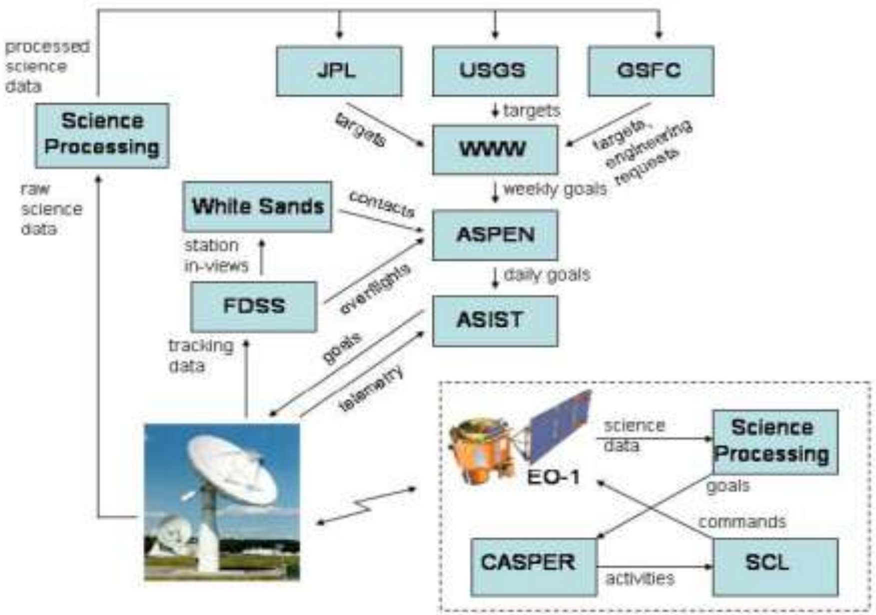
2. Technology:
   a. Planning and re-planning
   b. Automated data analysis

3. Successfully in operations:
   a. Operating onboard the Earth Observing-1 mission since 2003
   b. Transitioned to baseline and still operating
   c. Relevant example of mission cost reduction

ASE Architecture



Past Operations Flow



Current Operations Flow

[G. Rabideau, D. Tran, S. Chien, B. Cichy, R. Sherwood, D. Mandel, S. Frye, S. Shulman, J. Szwaxzkowski, D. Boyer, J. Van Gassbeck, IEEE International Conference on Space Mission Challenges for Information Technology. Pasadena, CA. July 2006]

GOAC

Goal Oriented Autonomous Controller

ESA – 2011

The Project

1. **Objective**: : A 2 years ESA study (2009-2011) to design and test **robust onboard goal-oriented autonomous* software**. The Goal Oriented Autonomous Controller (GOAC) is designed to generate plans in-situ, to deterministically dispatch activities for execution and to recover from off-nominal conditions.

2. **Approach**: The project aims at using state-of-the-art technologies deployed by different research centers to build a **control architecture suitable for challenging real-world scenarios at ESA**. Goal is to integrate recent AI and robotics research to cope with challenging requirements from space.

3. **Support**: The ESA P&S Architecture supports the GOAC architecture providing **fast planning and re-planning capabilities, plan dispatch and plan execution monitoring**

4. **Partners:**

5. **Project: ESA-ESTEC** ITTAO/1-5961/08/NL/RA

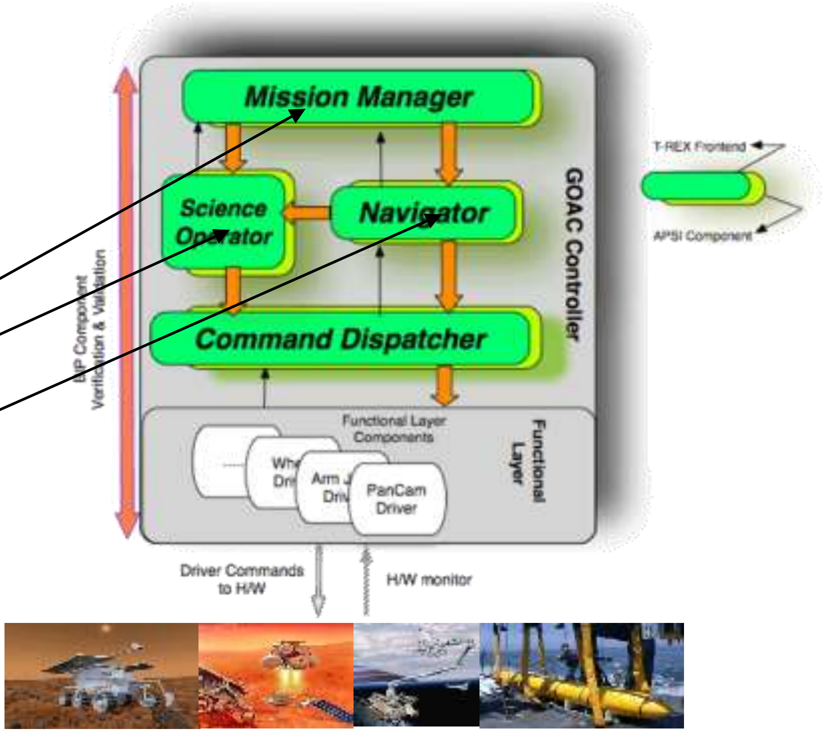
(*) Autonomy in this context is defined as "the ability to react to a wide variety of situations and circumstances, either nominal and exceptional, without the need of human supervision"

Approach

- An **executing agent** to control the synchronization and interaction of multiple control loops based on the sense-plan-act paradigm

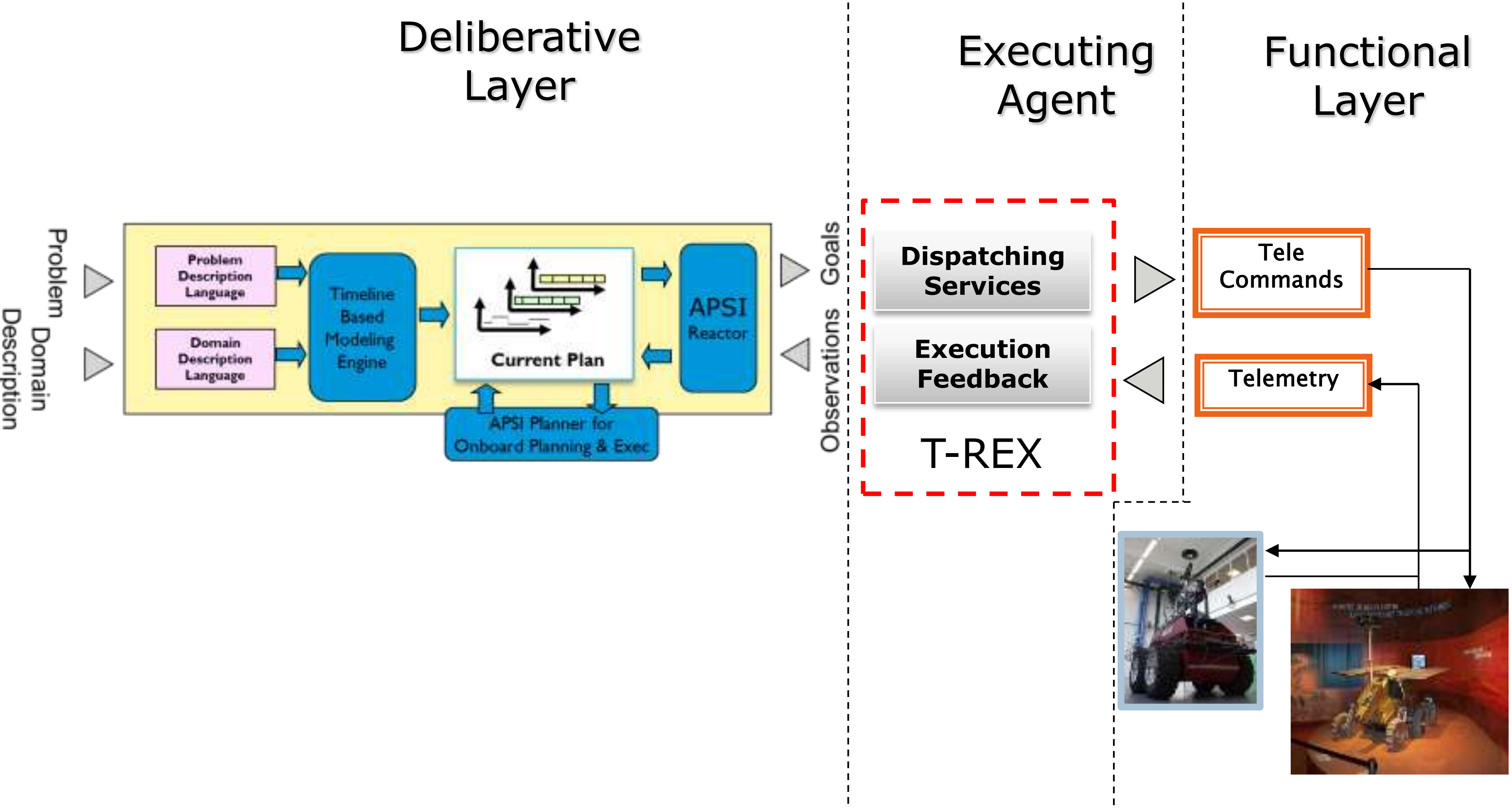
- **A set of APSI-Based deliberative reactors**, implementing the sense-plan-act paradigm as a P&S problem with timelines

- A functional layer to connect the architecture to HW platforms

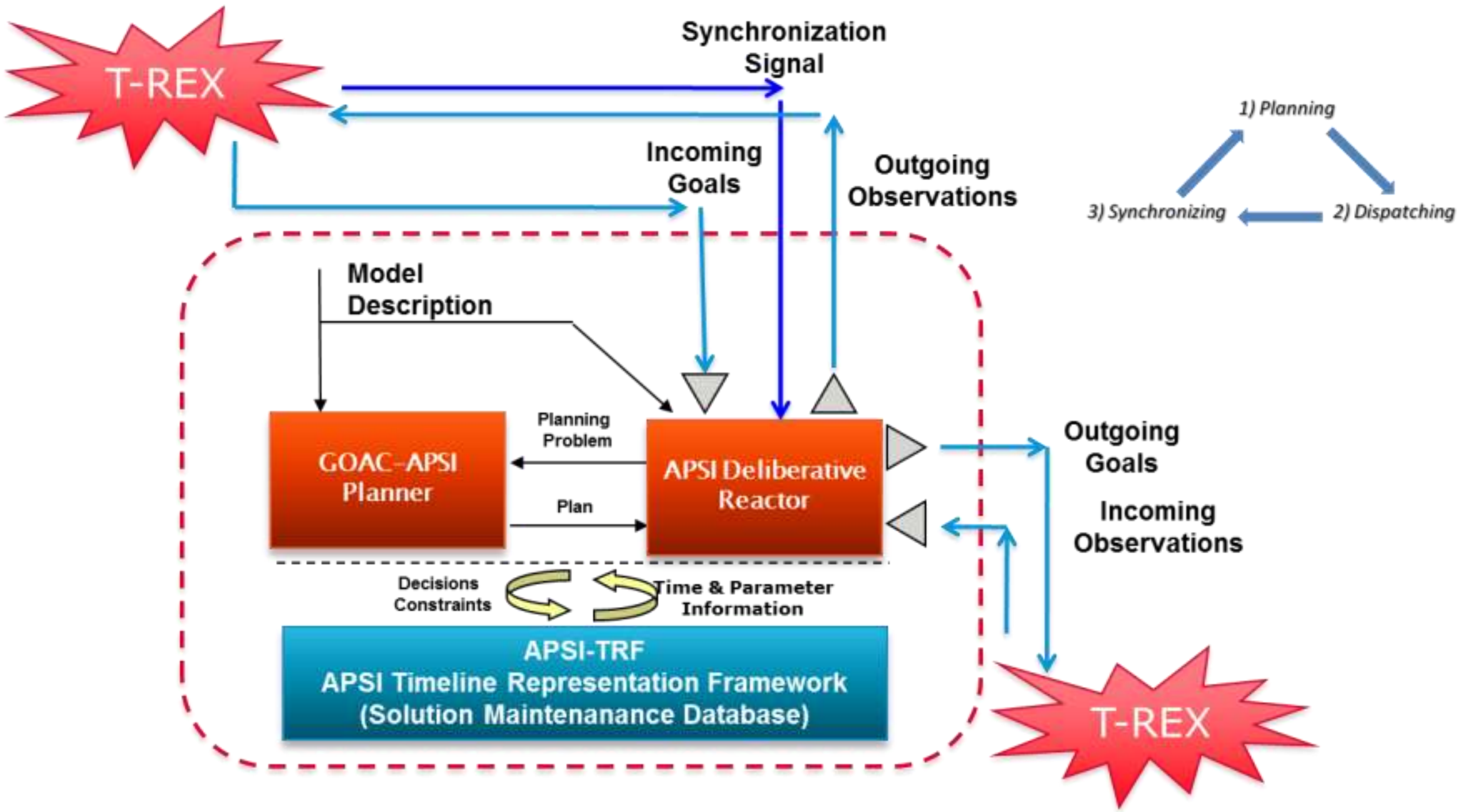


[F. Py, K. Rajan, and C. McGann. A systematic agent framework for situated autonomous systems. In AAMAS,, 2010.]

Conceptual Architecture

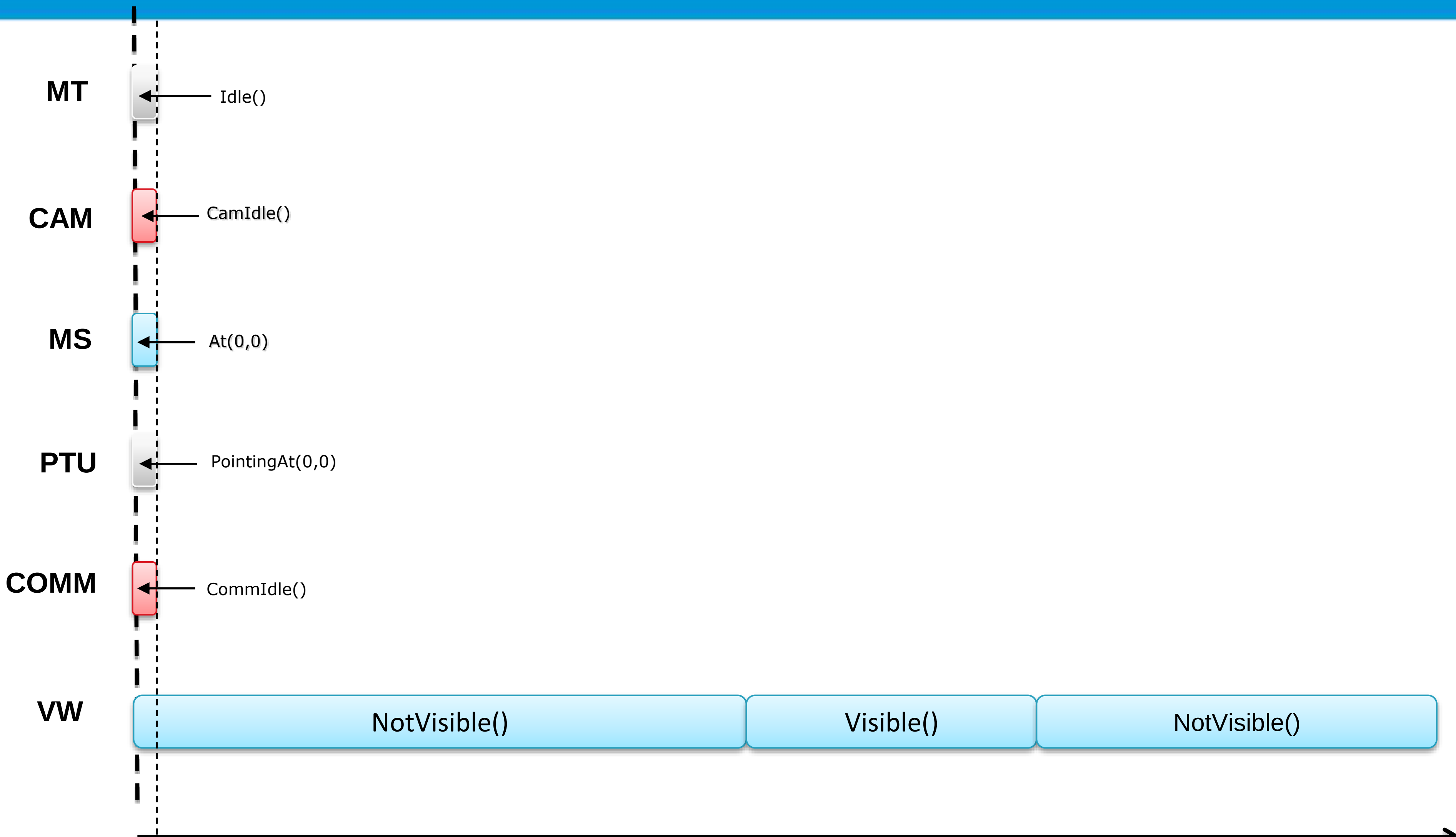


Deliberative Layer - Implementation



Initial Conditions

Execution…

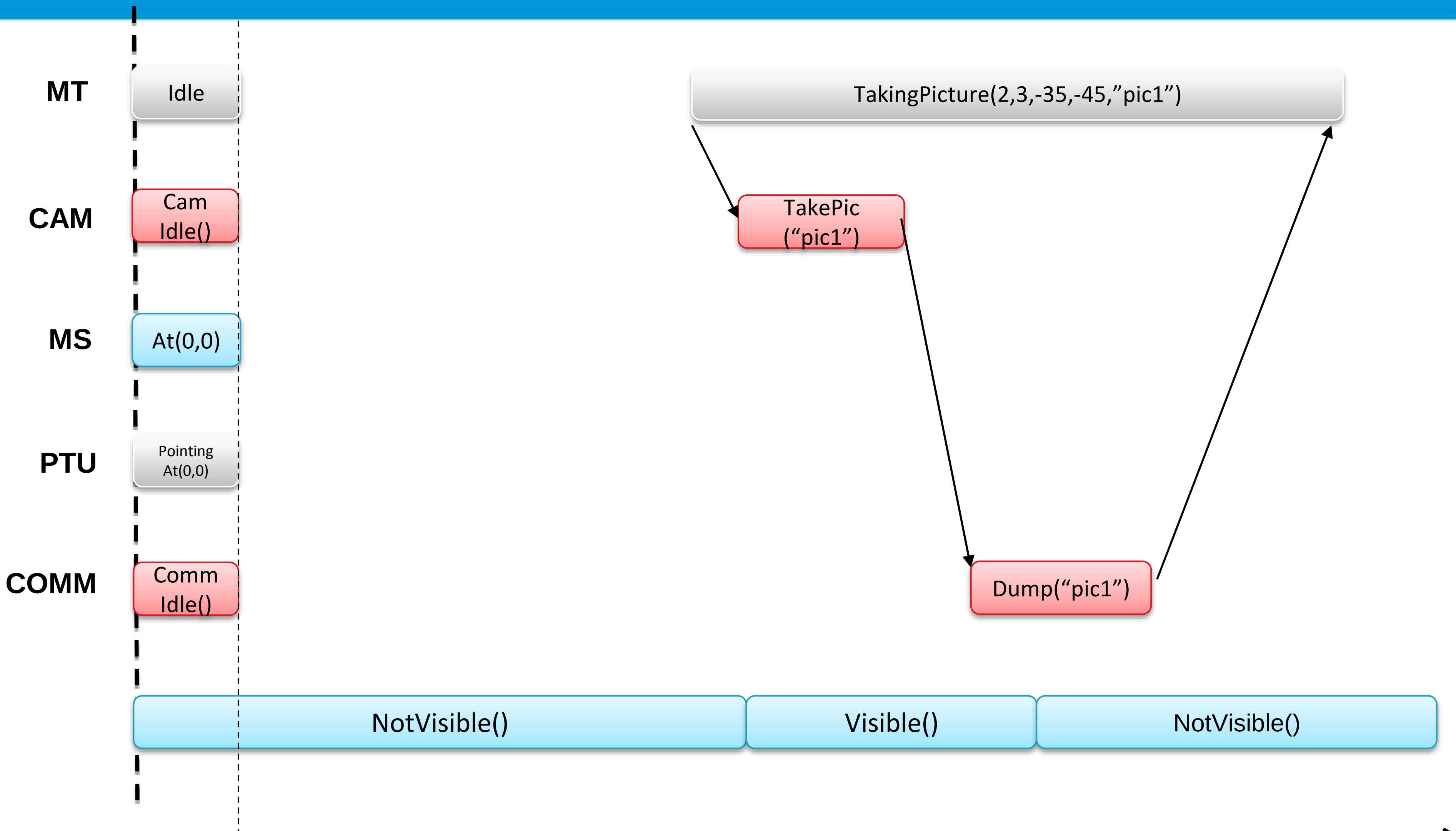

MT
Idle()
CAM
CamIdle()
MS
At(0,0)
PTU
PointingAt(0,0)
COMM
CommIdle()
VW
NotVisible()
Visible()
NotVisible()

Execution...

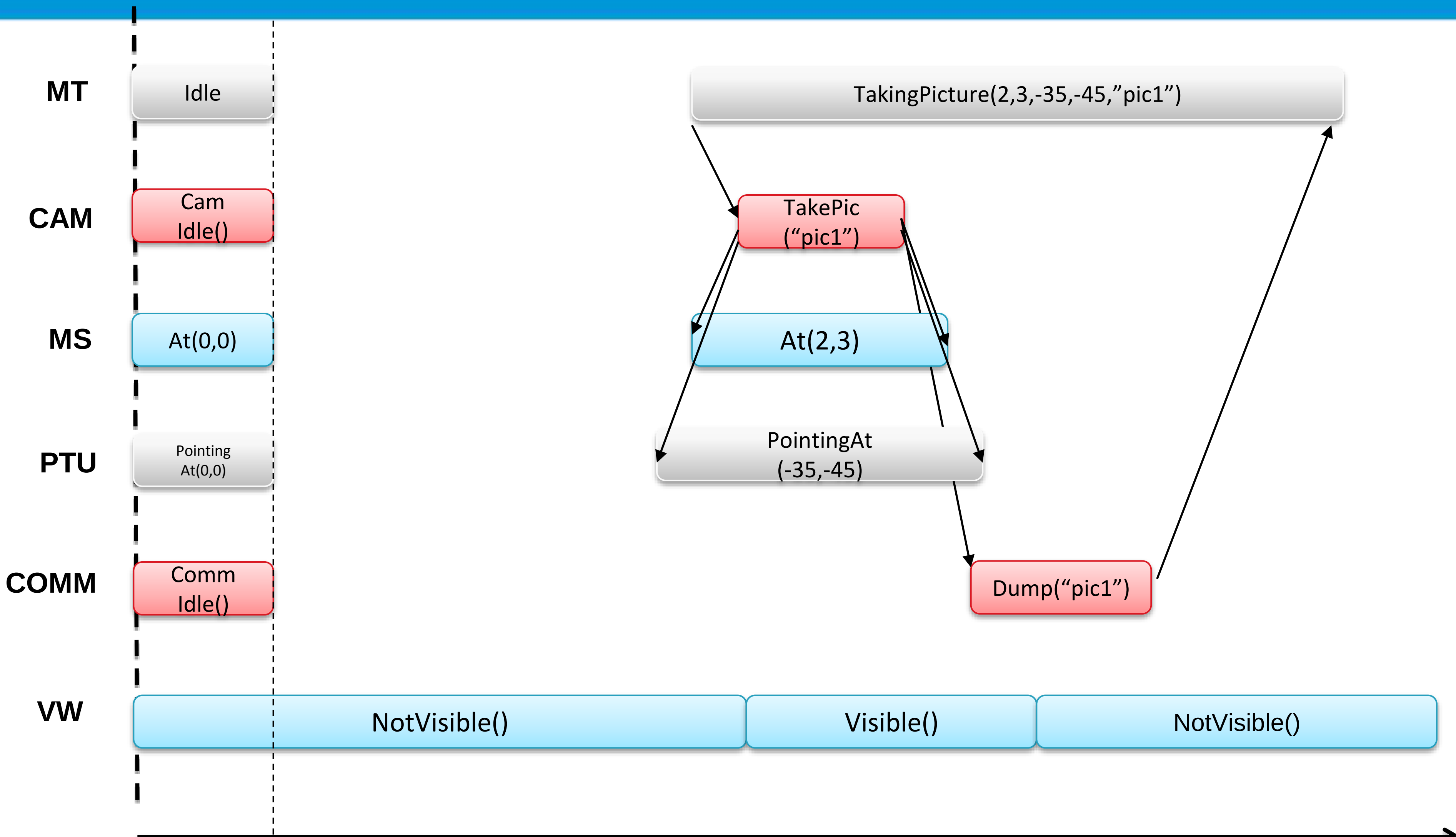
MT
Idle()
CAM
CamIdle()
MS
At(0,0)
PTU
PointingAt(0,0)
COMM
CommIdle()
NotVisible()
Visible()
NotVisible()

A goal coming in...

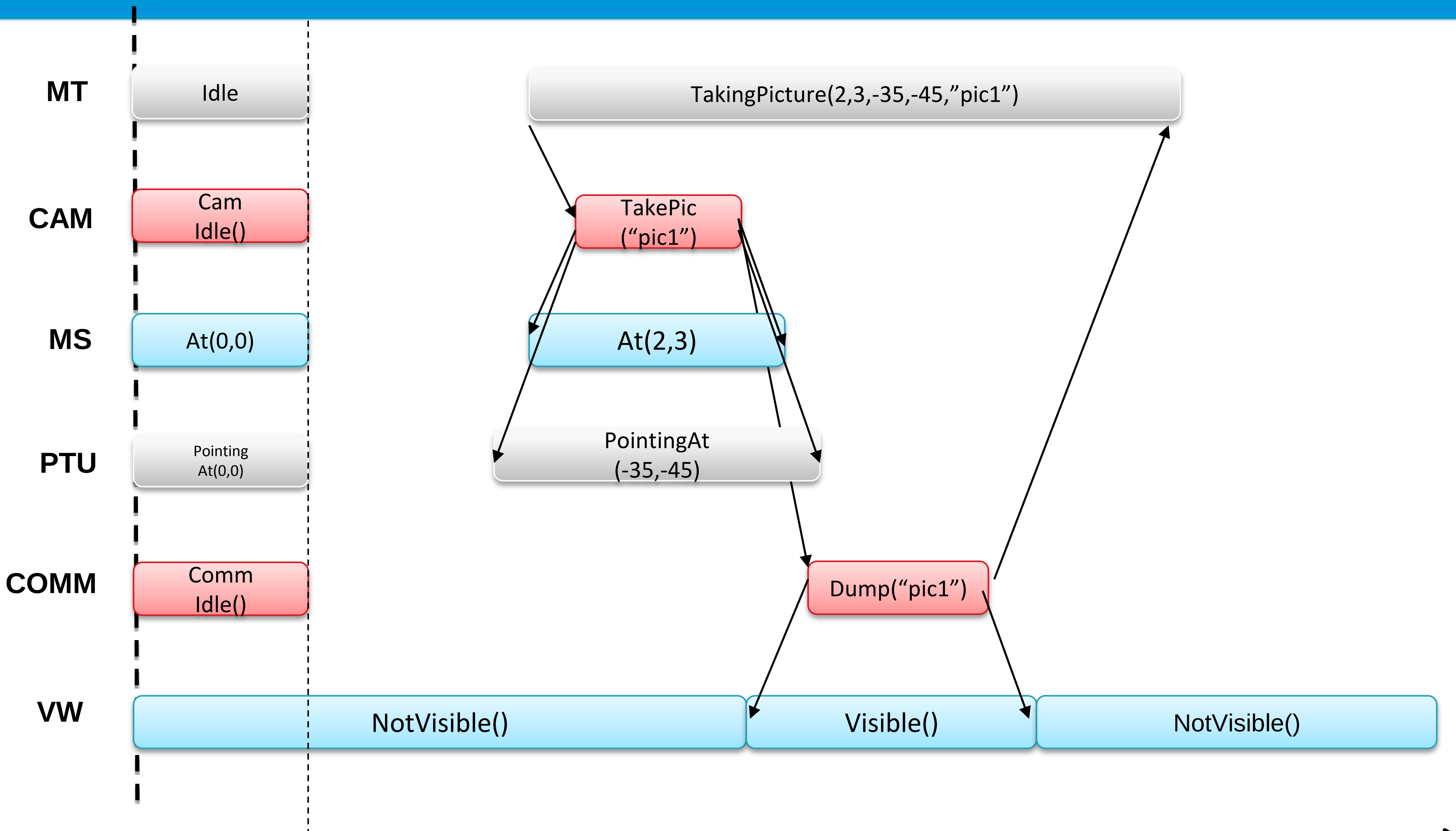
Planning...


MT
CAM
MS
PTU
COMM
Idle
Cam Idle()
At(0,0)
Pointing At(0,0)
Comm Idle()
TakingPicture(2,3,-35,-45,"pic1")
TakePic ("pic1")
Dump("pic1")
NotVisible()
Visible()
NotVisible()

Planning…

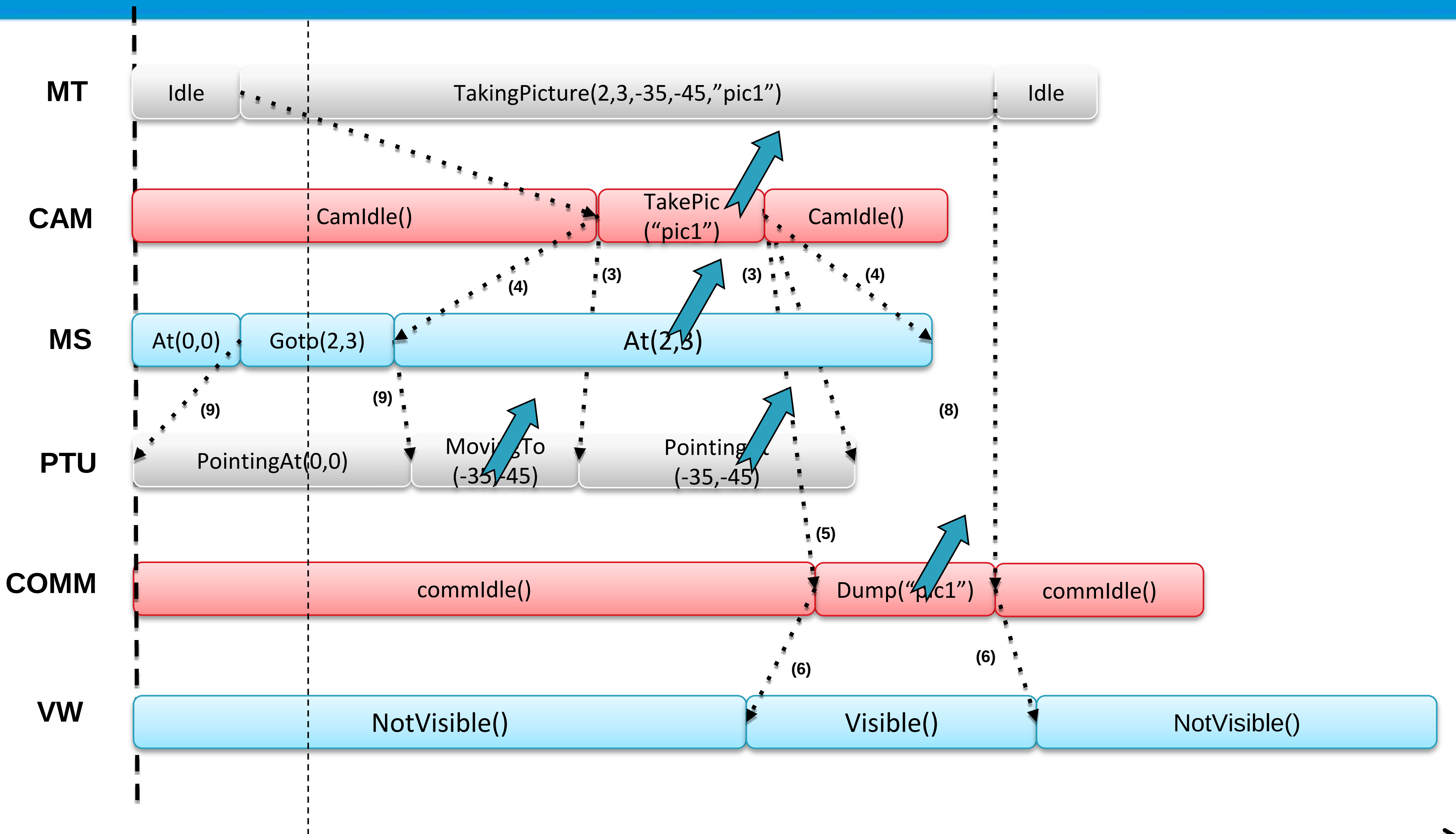

MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
CAM
Cam Idle()
TakePic ("pic1")
MS
At(0,0)
At(2,3)
PTU
Pointing At(0,0)
PointingAt (-35,-45)
COMM
Comm Idle()
Dump("pic1")
VW
NotVisible()
Visible()
NotVisible()

Planning…

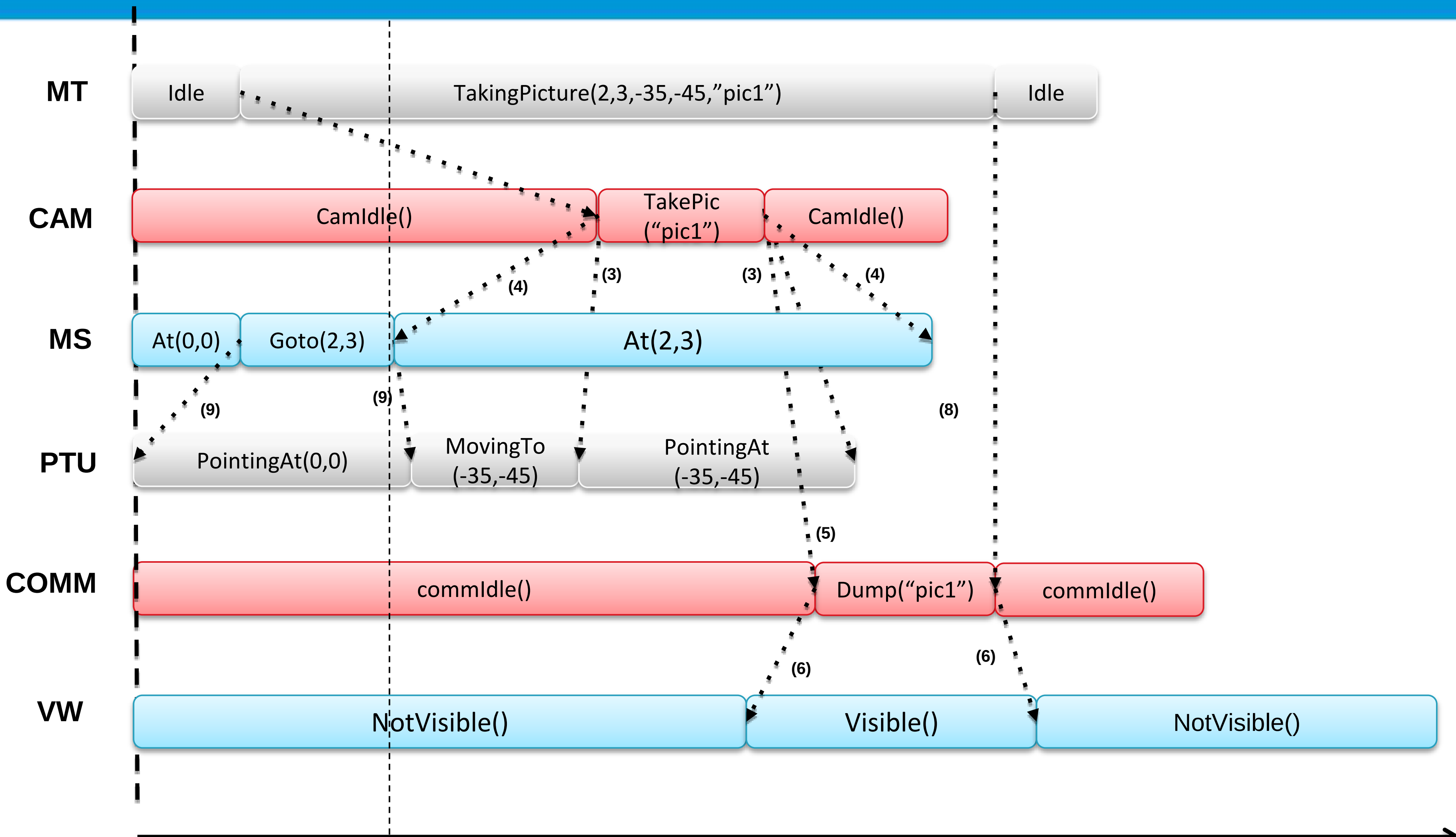


MT
CAM
MS
PTU
COMM
VW
Idle
TakingPicture(2,3,-35,-45,"pic1")
Cam
Idle()
TakePic
("pic1")
At(0,0)
At(2,3)
Pointing
At(0,0)
PointingAt
(-35,-45)
Comm
Idle()
Dump("pic1")
NotVisible()
Visible()
NotVisible()

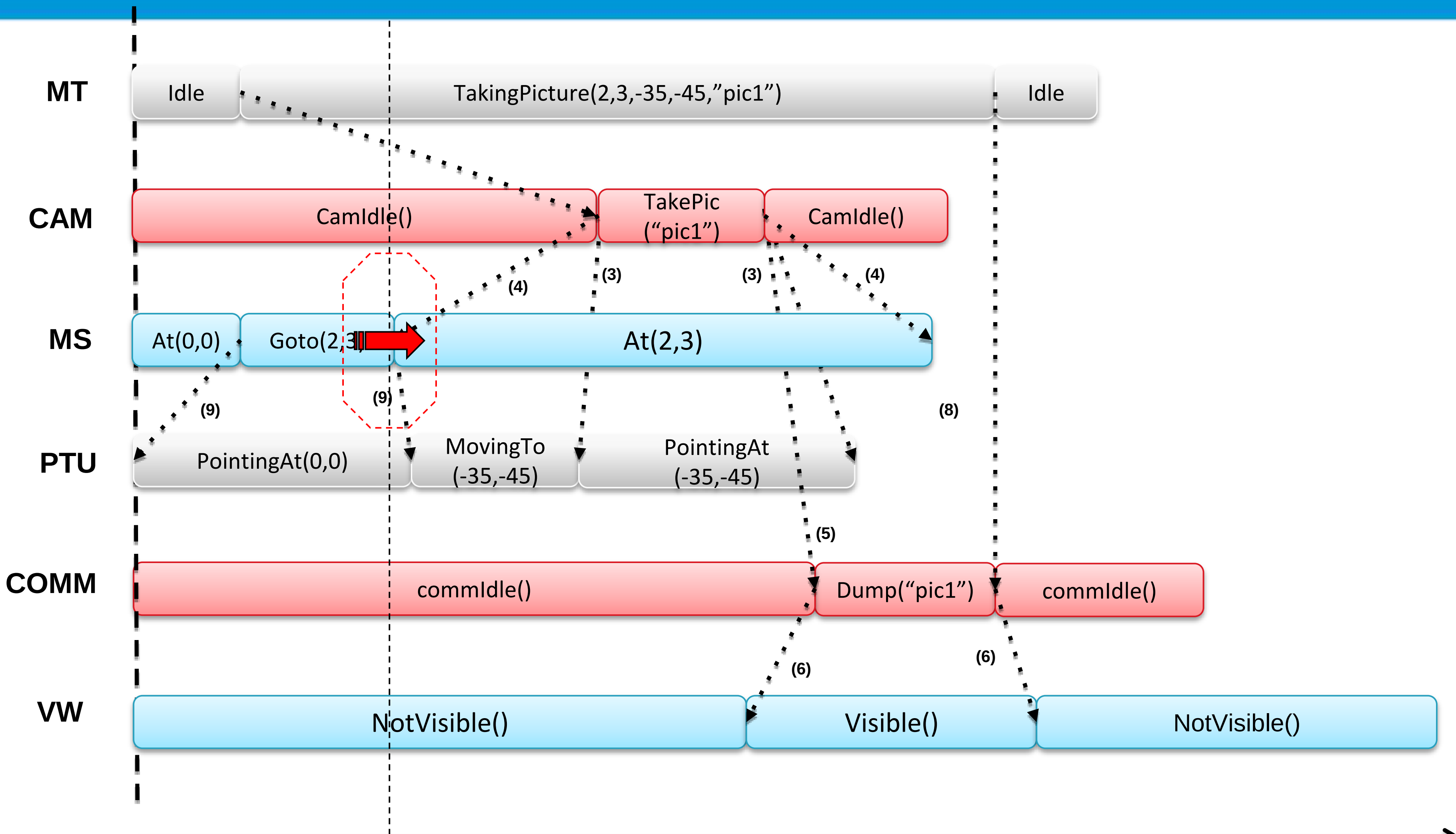
Dispatching and Executing…



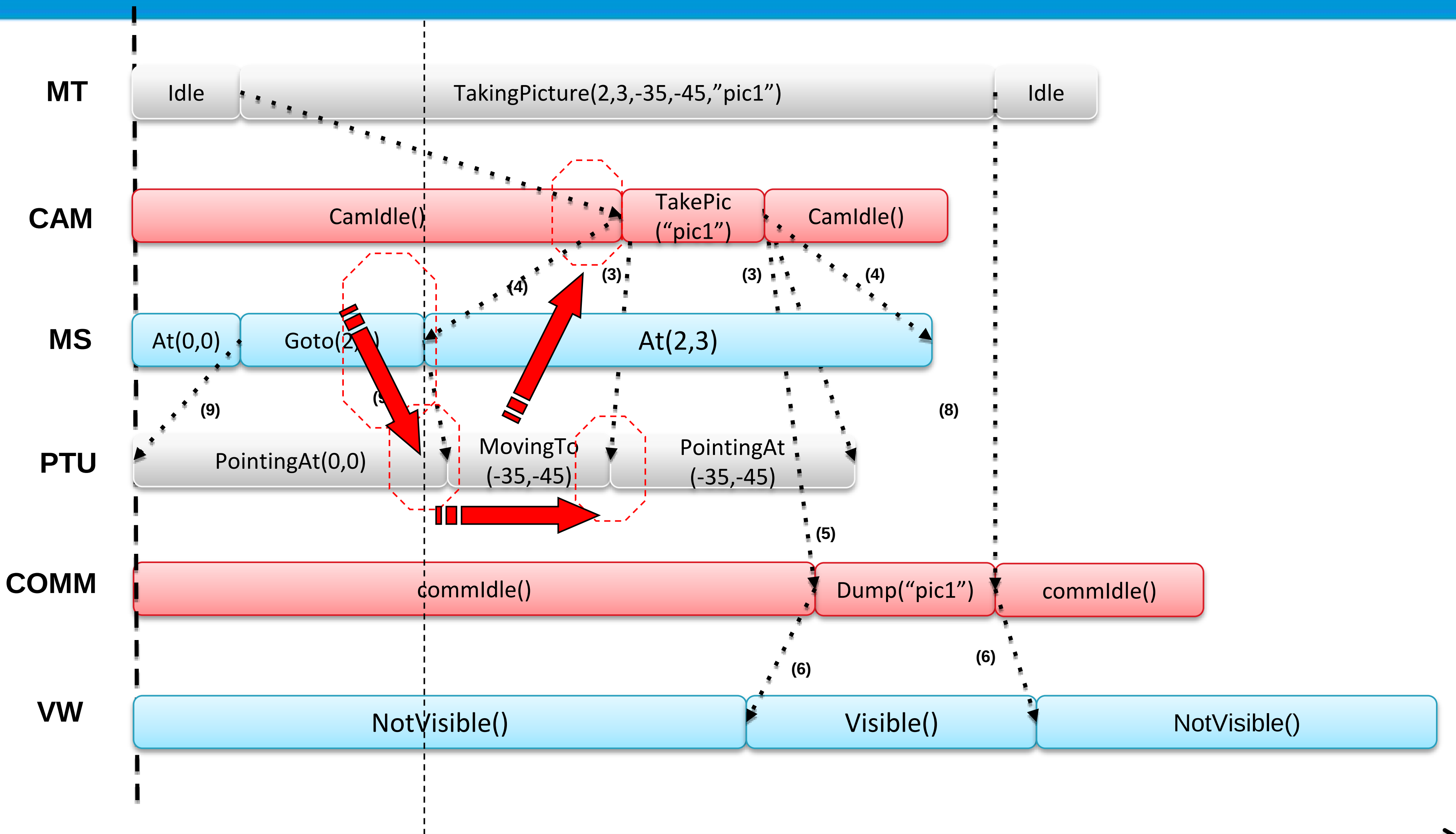
Executing...


MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
Idle
CAM
CamIdle()
TakePic ("pic1")
CamIdle()
(3)
(4)
(3)
(4)
MS
At(0,0)
Goto(2,3)
At(2,3)
(9)
(9)
(8)
PTU
PointingAt(0,0)
MovingTo (-35,-45)
PointingAt (-35,-45)
(5)
COMM
commIdle()
Dump("pic1")
commIdle()
(6)
(6)
VW
NotVisible()
Visible()
NotVisible()

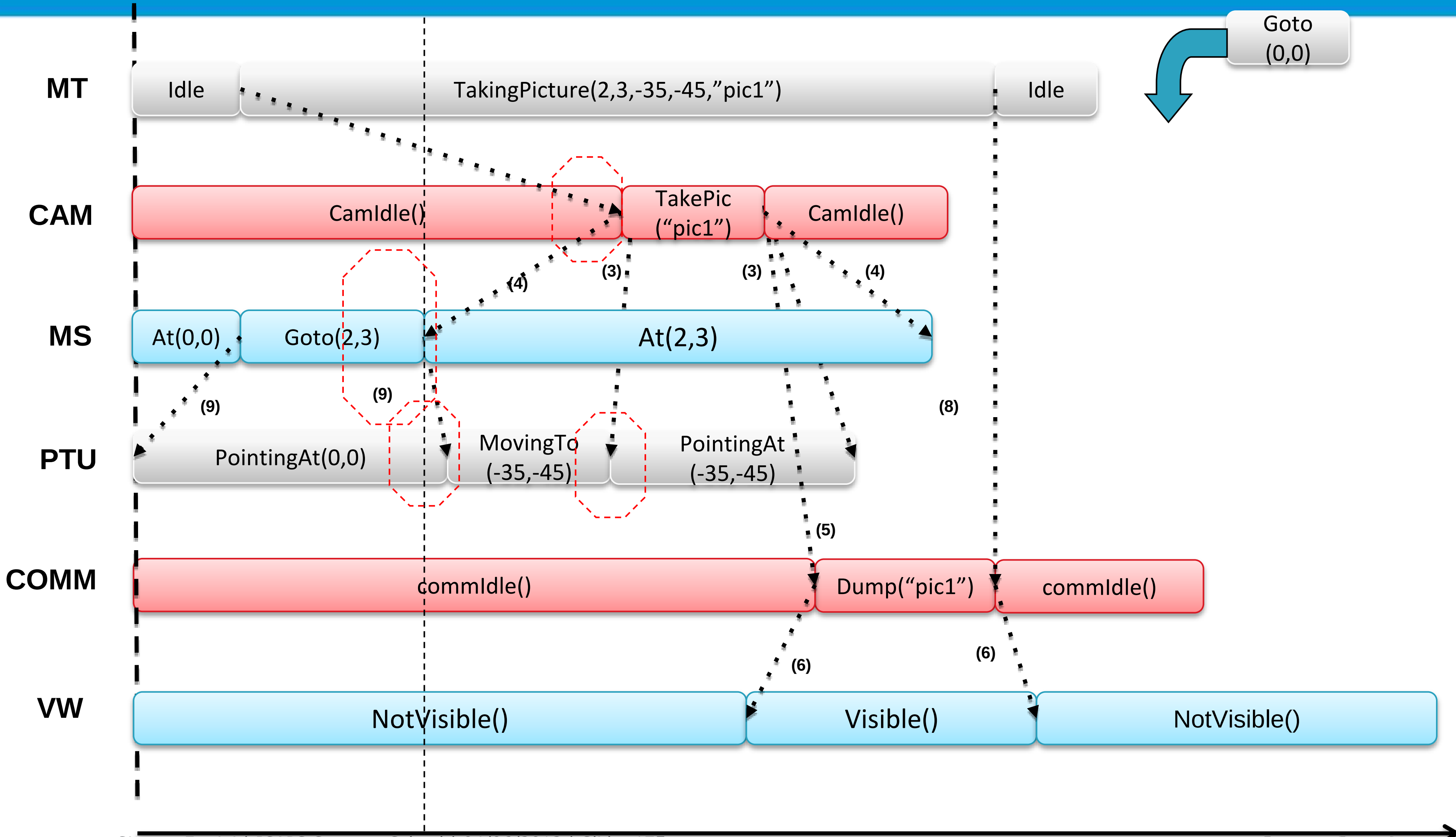
Synchronize…



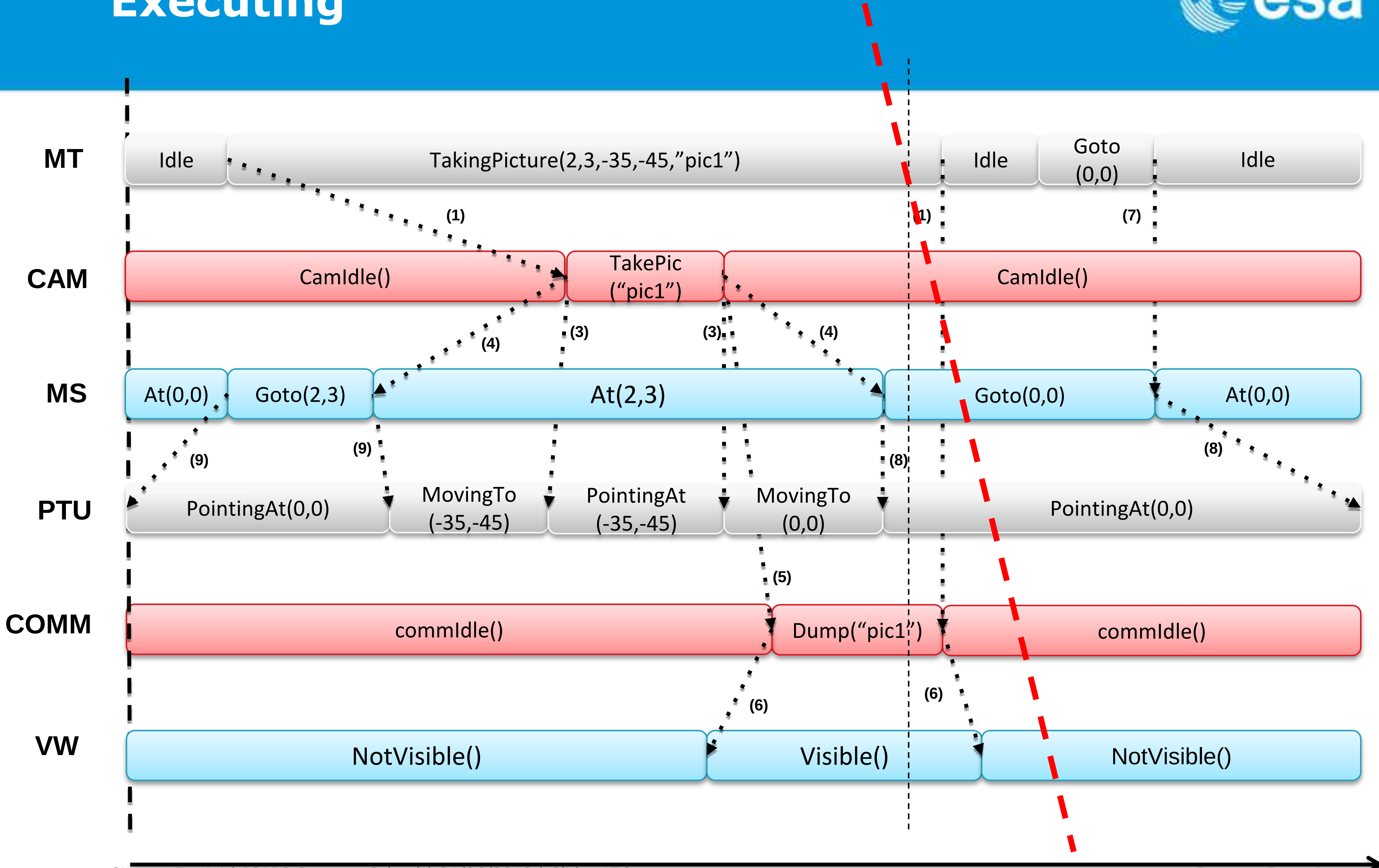
Synchronize…



Synchronize and Plan…

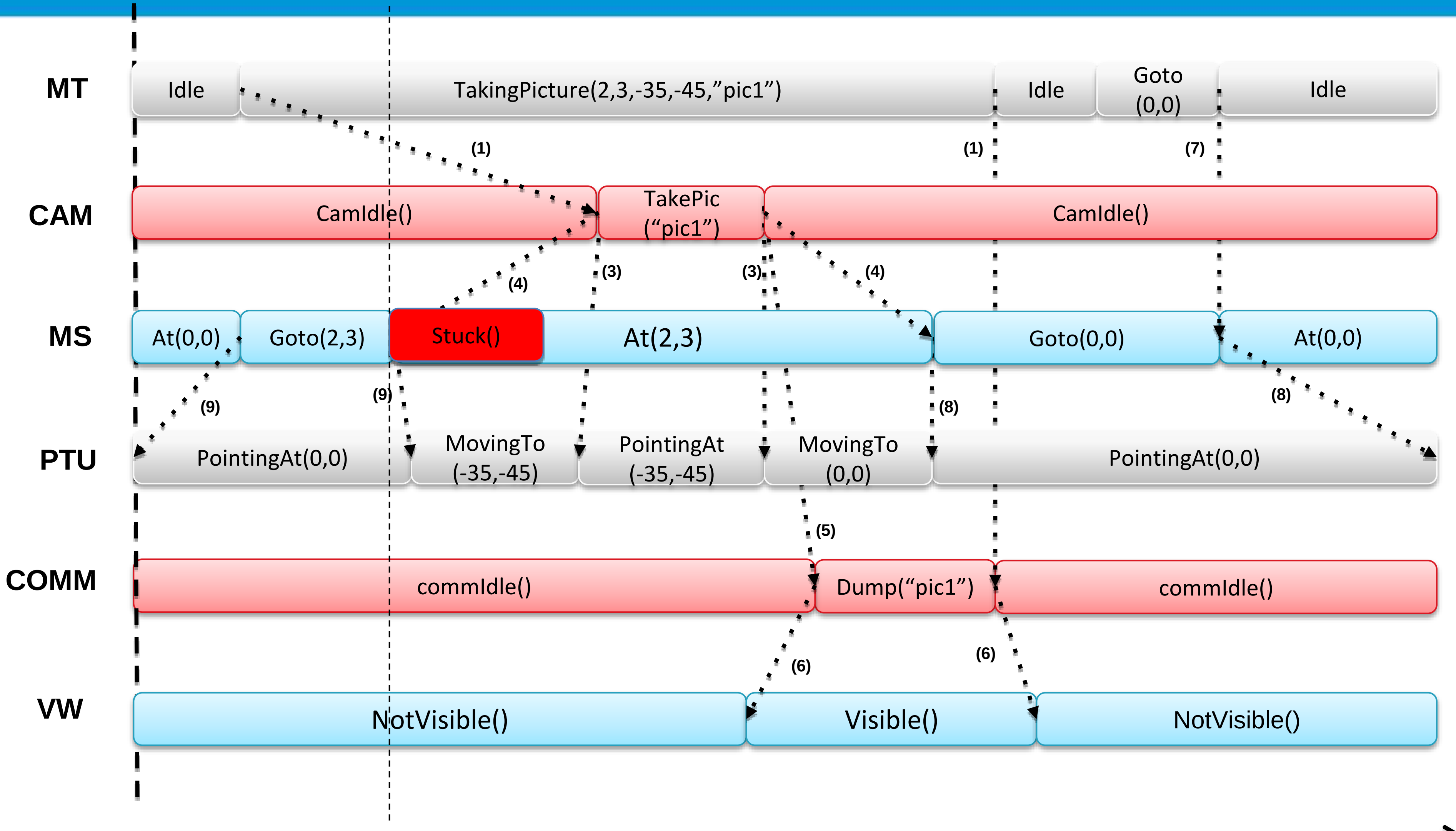


Executing



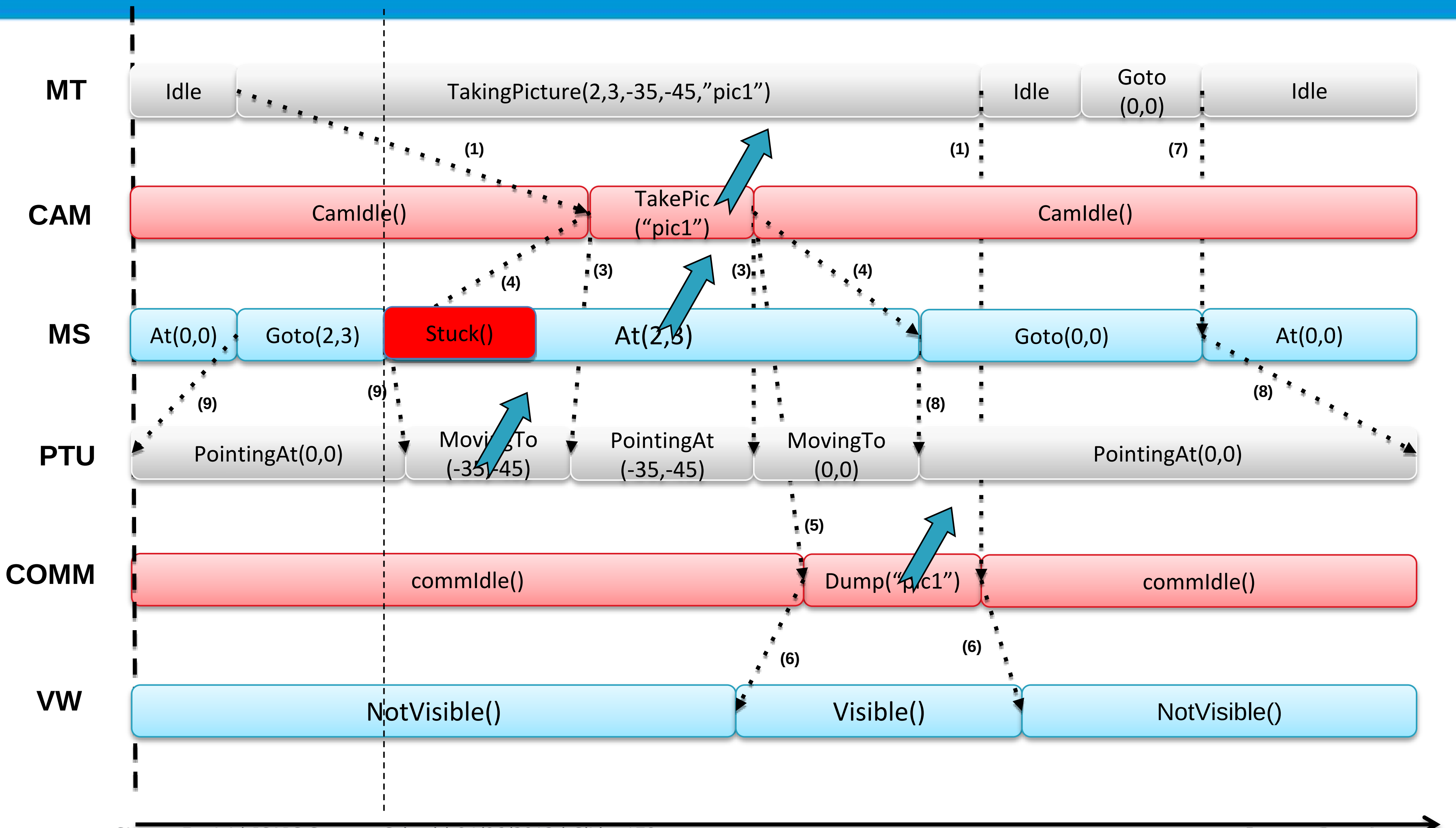
MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
Idle
Goto (0,0)
Idle
(1)
(7)
CAM
CamIdle()
TakePic ("pic1")
CamIdle()
(3)
(3)
(4)
(4)
MS
At(0,0)
Goto(2,3)
At(2,3)
Goto(0,0)
At(0,0)
(9)
(9)
(8)
(8)
PTU
PointingAt(0,0)
MovingTo (-35,-45)
PointingAt (-35,-45)
MovingTo (0,0)
PointingAt(0,0)
(5)
COMM
commIdle()
Dump("pic1")
commIdle()
(6)
(6)
VW
NotVisible()
Visible()
NotVisible()

Failure!



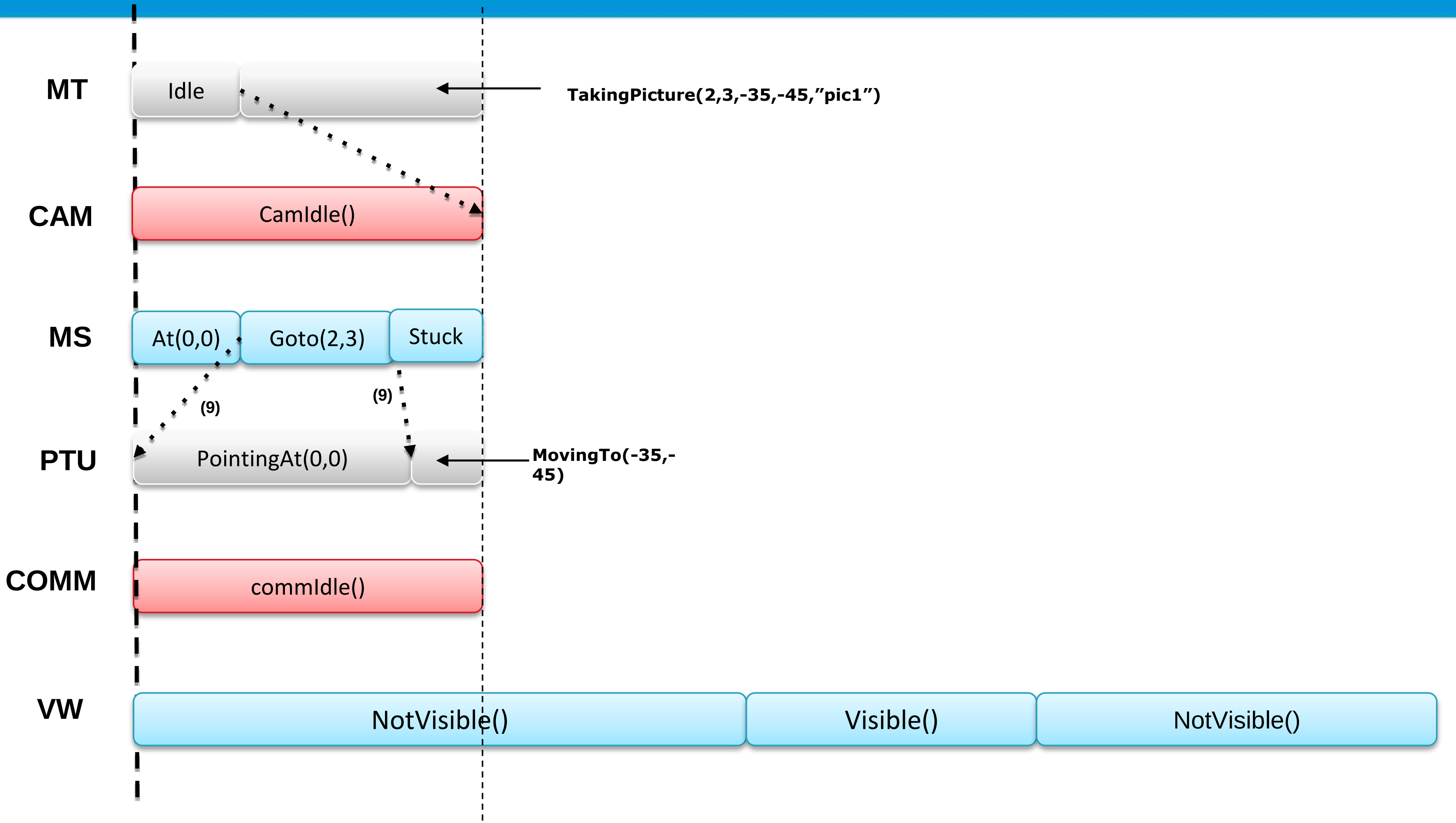
MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
Idle
Goto (0,0)
Idle
(1)
(1)
(7)
CAM
CamIdle()
TakePic ("pic1")
CamIdle()
(3)
(3)
(4)
(4)
MS
At(0,0)
Goto(2,3)
Stuck()
At(2,3)
Goto(0,0)
At(0,0)
(9)
(9)
(8)
(8)
PTU
PointingAt(0,0)
MovingTo (-35,-45)
PointingAt (-35,-45)
MovingTo (0,0)
PointingAt(0,0)
(5)
COMM
commIdle()
Dump("pic1")
commIdle()
(6)
(6)
VW
NotVisible()
Visible()
NotVisible()

Recall!

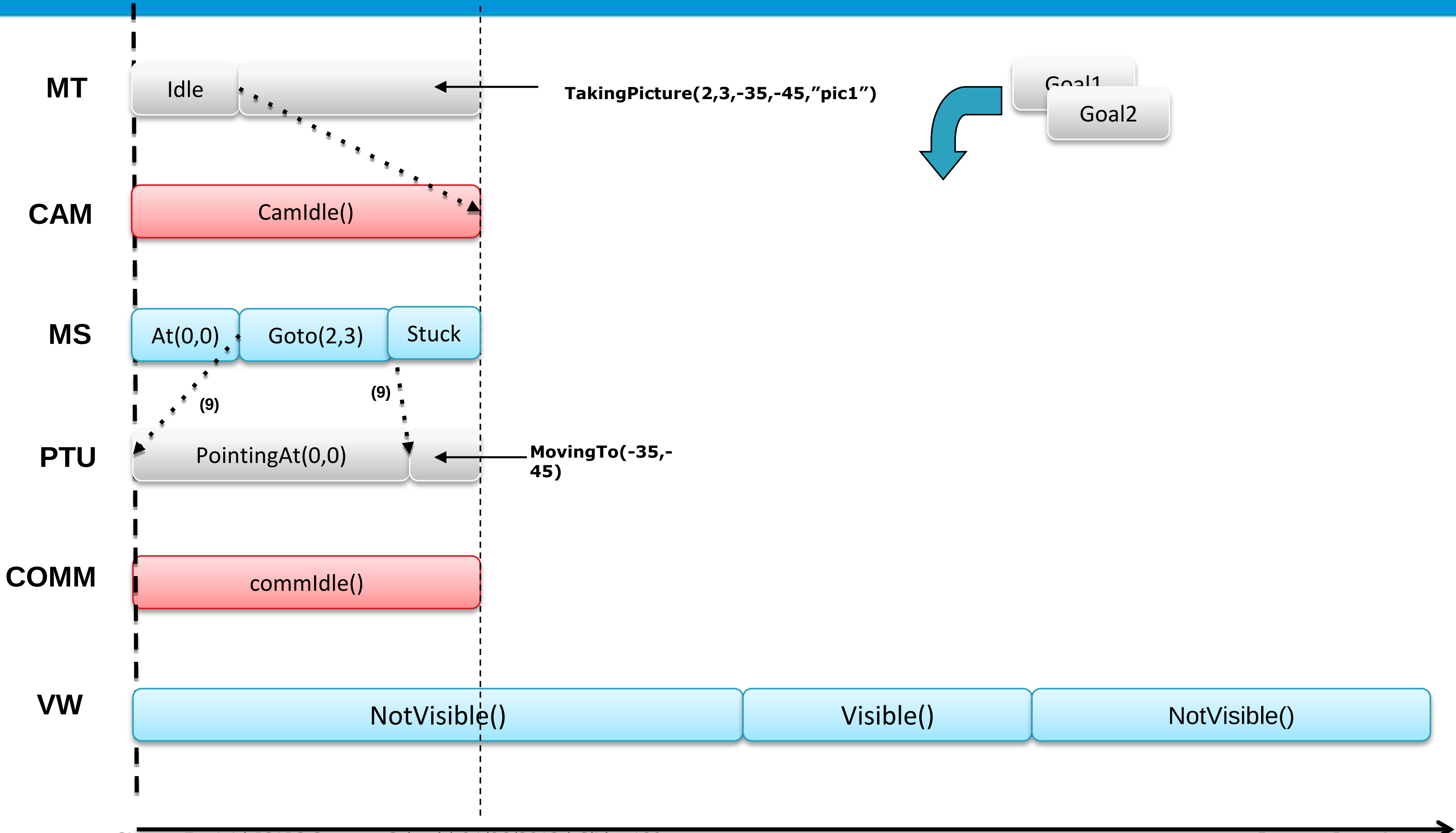


MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
Idle
Goto (0,0)
Idle
(1)
(1)
(7)
CAM
CamIdle()
TakePic ("pic1")
CamIdle()
(4)
(3)
(3)
(4)
MS
At(0,0)
Goto(2,3)
Stuck()
At(2,3)
Goto(0,0)
At(0,0)
(9)
(9)
(8)
(8)
PTU
PointingAt(0,0)
MovingTo (-35,-45)
PointingAt (-35,-45)
MovingTo (0,0)
PointingAt(0,0)
(5)
COMM
commIdle()
Dump("pic1")
commIdle()
(6)
(6)
VW
NotVisible()
Visible()
NotVisible()

Current Status



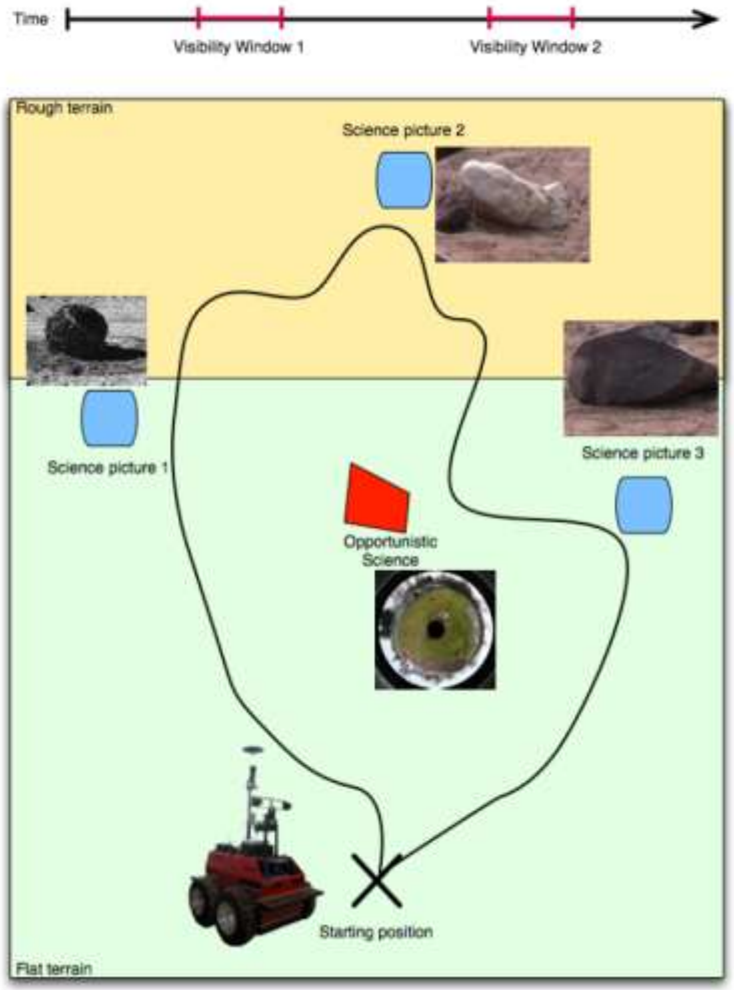
Re-Planning



MT
Idle
TakingPicture(2,3,-35,-45,"pic1")
Goal1
Goal2
CAM
CamIdle()
MS
At(0,0)
Goto(2,3)
Stuck
(9)
(9)
PTU
PointingAt(0,0)
MovingTo(-35,-45)
COMM
commIdle()
VW
NotVisible()
Visible()
NotVisible()

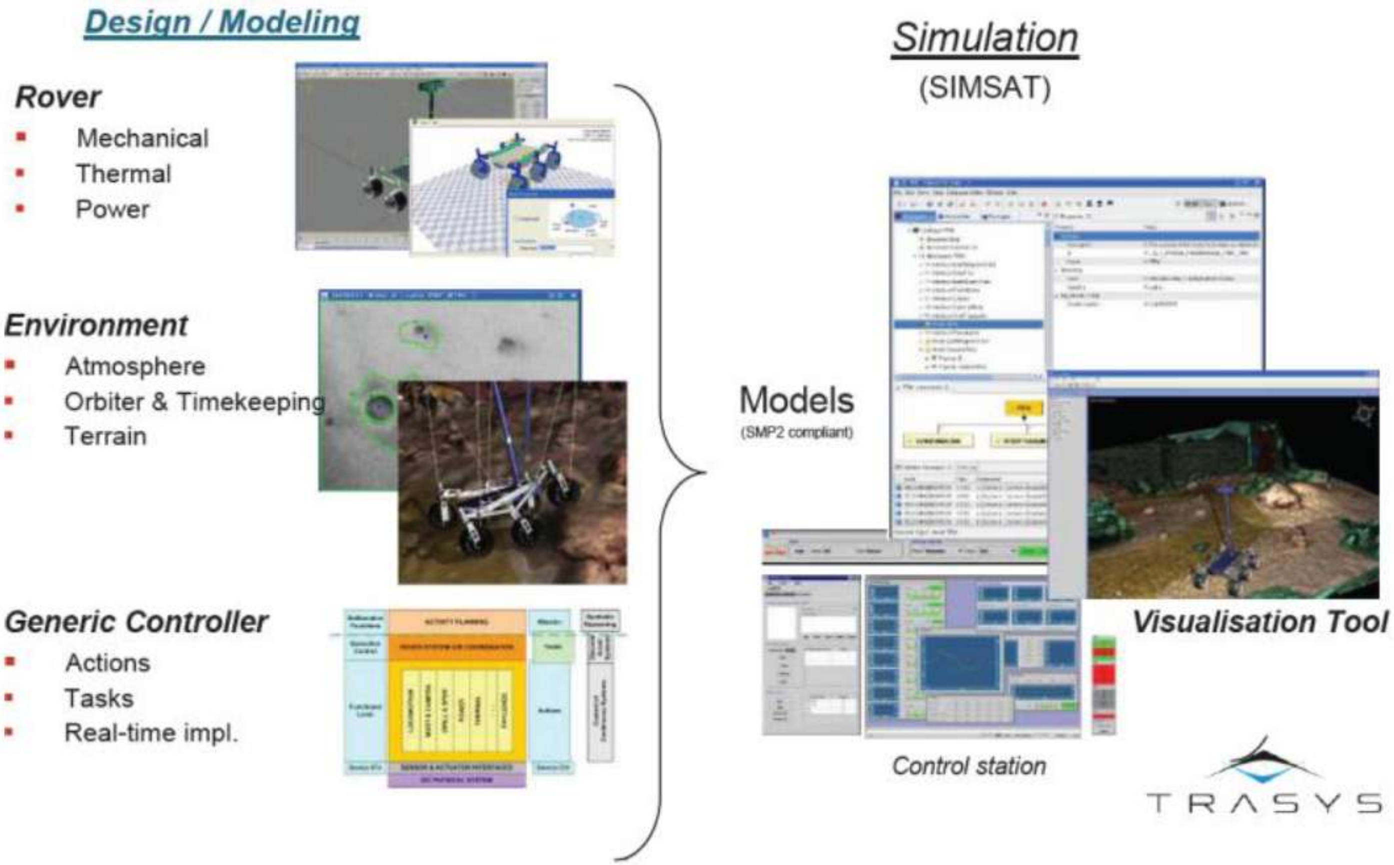
CASE STUDY 1 - Scenario

- Navigate safely in an a priori unknown environment.
- Take high resolution pictures of an operator-given list of locations (of scientific interest).
- Communicate with orbiters or a lander during some given visibility windows.
- Continuously monitor the environment for "opportunistic science," and take appropriate actions when something interesting is detected.
- Monitor and control the proper heating of the platform and the payload.
- Monitor and control the proper power usage and energy consumption.



CASE STUDY 2 – 3DROV

- Planetary exploration simulation environment to support:
  - Early system design and technical assessment considering "rover-in-context"
  - Specific engineering studies (e.g. mobility, autonomy, operations)
  - Early insight to rover operations



Slide from GMV

References

- *General References on CSP*

    - Montanari, U. (1974). Networks of Constraints: Fundamental Properties and Applications to Picture Processing. Information Sciences, 7 , 95–132.
    - Dechter, R., Meiri, I., Pearl, J., Temporal constraint networks. Artificial Intelligence 49 (1991) 61–95

- *Dynamic temporal networks*

    - Cesta, A., Oddi, A., Gaining Efficiency and Flexibility in the Simple Temporal Problem, in L. Chittaro, S. Goodwin, H. Hamilton, A.Montanari, Proc. 3rd Int. Workshop on Temporal Representation and Reasoning (TIME-96), Key West, FL, May 19-20, 1996, IEEE Computer Society Press:May 1996
    - Cesta A. and Oddi, A. “Algorithms for Dynamic Management of Temporal Constraints Networks”. Technical Reports 2001.
    - Boris V. Cherkassky, Andrew V. Goldberg, Negative-Cycle Detection Algorithms, European Symposium on Algorithms, 1996
    - Demetrescu C, Emiliozzi S., Italiano G. F., Experimental analysis of dynamic all pairs shortest path algorithms, SODA 2004: 369-378
    - A. Oddi and A.Cesta. Incremental Forward Checking for the Disjunctive Temporal Problem. W.Horn (Ed.): ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence, IOS Press: Amsterdam, 2000, pp. 108-112.

References

- *Planning, Scheduling & Execution with Timelines*

  - Muscettola, N. 1994. HSTS: Integrating Planning and Scheduling. In Zweben, M. and Fox, M.S., ed., Intelligent Scheduling. Morgan Kauffmann.
  - Frank, J., and Jonsson, A. 2003. Constraint-based attribute and interval planning. Constraints 8(4):339–364.
  - A. Cesta, S. Fratini and A. Oddi. Planning with Concurrency, Time and Resources: A CSP-Based Approach. In I. Vlahavas, D. Vrakas (Eds.), "Intelligent Techniques for Planning", Idea Group Publishing, pp. 259-295, 2005
  - Fratini, S., Pecora, F., and Cesta, A. Unifying Planning and scheduling as Timelines in a Component-Based Perspective. In Archives of Control Sciences, 18(2):231-271, 2008
  - Cédric Pralet, Gérard Verfaillie: Using Constraint Networks on Timelines to Model and Solve Planning and Scheduling Problems. ICAPS 2008: 272-279
  - Cesta, A., and Fratini, S. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment, In PlanSIG-08. Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group, Edinburgh, UK, December, 2008.
  - Fratini, S.; Pecora, F.; and Cesta, A. 2008. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. Archives of Control Sciences 18(2):231–271
  - Verfaillie, G.; Pralet, C.; and Lematre, M. 2010. How to model planning and scheduling problems using constraint networks on timelines. Knowledge Eng. Review 25(3):319–336
  - Fratini, S., Cesta, A. De Benedictis, R., Orlandini, A., and Rasconi, R. APSI-Based Deliberation in Goal Oriente Autonomous Controllers. In ASTRA-11. Proceedings of the 11th Workshop on Advanced Space Technologies for Robotics and Automation, ESA-ESTEC, April, 2011
  - Fratini, S., and Cesta, A. 2012. The APSI Framework: A Platform for Timeline Synthesis. In Proceedings of the 1st Workshops on Planning and Scheduling with Timelines at ICAPS-12, Atibaia, Brazil
  - Frank, J. What is a Timeline? In Proceedings of the 4th Workshops on Knowledge Engineering for Planning and Scheduling at ICAPS-13, Rome, Italy

References

- *Space Applications*

  - N. Muscettola, S.F. Smith, A. Cesta, and D. D'Aloisi. Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Architecture. In IEEE Control Systems, 12(1):28-37, 1992
  - Muscettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote agent: to go boldly where no AI system has gone before. Artificial Intelligence, 103(1–2):5–48.
  - S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran, "ASPEN - Automated Planning and Scheduling for Space Mission Operations", In Proceedings of SpaceOps'00, 2000.
  - Cesta, A., Cortellessa, G., Denis, M., Donati, A., Fratini, S., Oddi, A., Policella, N., Rabenau, E. and Schulster, J. MEXAR2: AI Solves Mission Planner Problems. IEEE Intelligent Systems, 22(4):12-19, July/August, 2007
  - Cesta, A., Cortellessa, G., Fratini, S., Oddi, A. and Policella. N. An Innovative Product for Space Mission Planning -- an a posteriori evaluation. In ICAPS-07. Proceedings of the 17th International Conference on Automated Planning and Scheduling, 2007
  - Cesta, A., Cortellessa, G., Fratini, S., Oddi, A. Developing an End-to-En d Planning Application from a Timeline Representation Framework, IAAI-09, 21st Innovative Application of Artificial Intelligence Conference, Pasadena, CA, July, 2009.
  - Cesta, A., Cortellessa, G., Fratini, S., and Oddi, A., Bernardi, G., Deploying Interactive Mission Planning Tools – Experiences and Lessons Learned. Journal of Advanced Computational Intelligence and Intelligent Informatics. Vol.15 No.8, pp. 1149-1158, 2011
  - Chien, S., Johnston, M., Frank, J., Giuliano, M., Kavelaars, A., Lenzen, C., Policella, N. A Generalized Timeline Representation, Services, and Interface for Automating Space Mission Operations. Proceedings of the 12th International Conference on Space Operations, SpaceOps 2012.