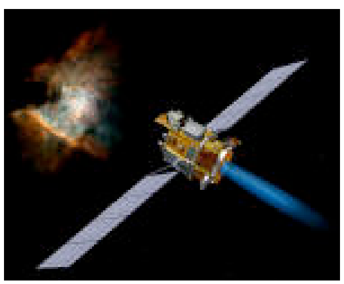
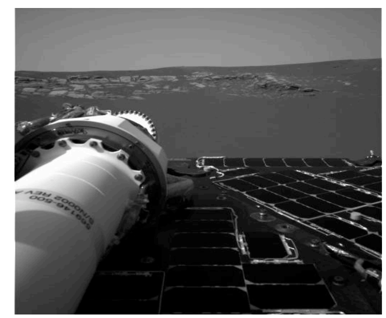
Ames Research Center

Constraint Reasoning in Zero Gravity





Jeremy Frank

NASA Ames Research Center

Moffett Field, CA

Ames Research Center

Outline

- Classical Constraint Reasoning
  - "Simple" problems
- NASA Applications
  - "Complex" problems
- NASA Technology
  - Pushing Constraint Reasoning
- Open Research Areas: A Challenge

Ames Research Center

Classical Constraint Reasoning

- Binary CSPs
  - scope=2, bit-matrix representation of constraints
- Static CSPs
  - Solve fixed problem
- "Homogeneous" constraints
  - Binary constraint matrices
  - Interval reasoning, Temporal reasoning
  - A few on DEs, real-valued functions, heterogeneous
- Application integration considerations
  - "Infinite" resources to solve CSP problems
  - Stand-alone systems

Ames Research Center

Classical Constraint Reasoning

- CP 2004 papers (full-length)
  - Binary CSPs: 7
    - 1 of these is counting rather than satisfaction
  - AllDiff CSPs: 2
  - Linear constraints: 4
    - 3 of these have optimization criteria
  - SAT: 3
  - MAXSAT: 2
  - Quantified CSP/Quantified SAT: 3
  - Consistency of single constraint class: 7
  - Set Constraints: 3
  - Portfolio optimization:1

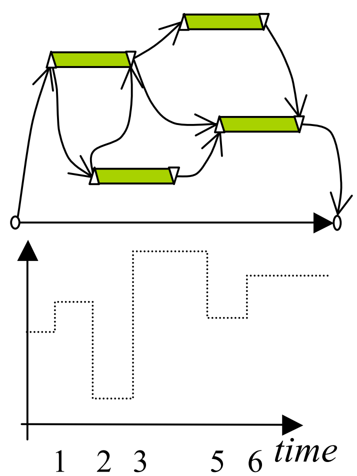
Ames Research Center

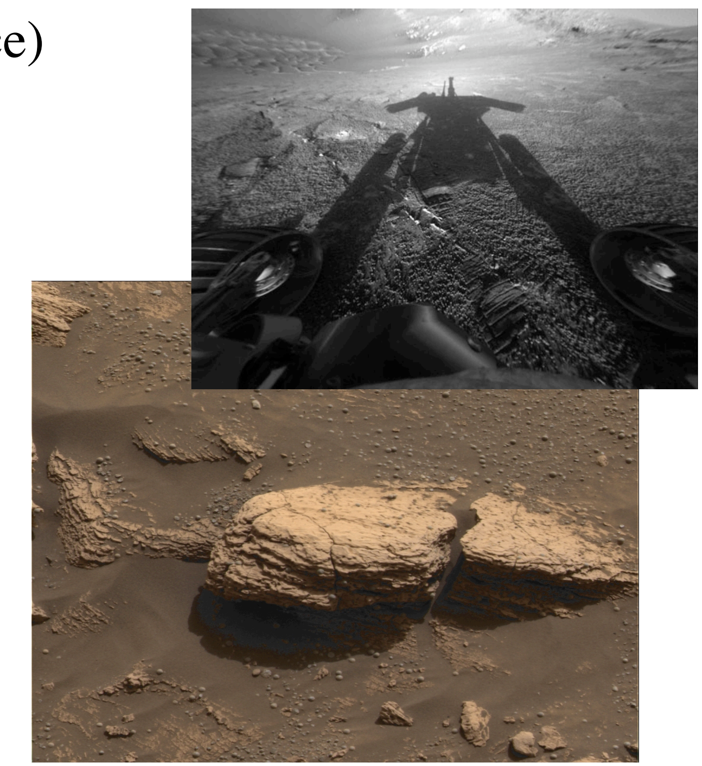
Classical Constraint Reasoning

- CP 2004 papers (full-length)
  - Local search algorithms: 3
    - applicable to heterogeneous problems
  - Global search algorithms: 1
    - applicable to heterogeneous problems
  - Heterogeneous constraints (non-scheduling): 2
  - Time + resource constraints: 6
    - 2 of these have optimization criteria
- What's missing this year? (but has been work in the past)
  - Dynamic CSPs
  - "Complex" constraints, e.g. DEs
  - Uncertainty
  - Integration story

NASA Applications

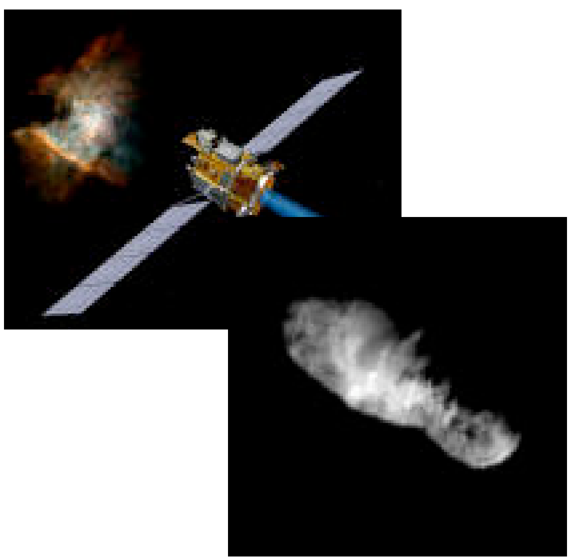
- Constraint Reasoning used in Missions
  - MER - Mars Exploration Rover Science Planning Tool ‘03-04
  - Life in the Artacama (LITA) Desert Rover ‘04
    - Onboard memory (renewable resource)
    - Power
    - Route planning
    - Causal constraints (AI Planning)

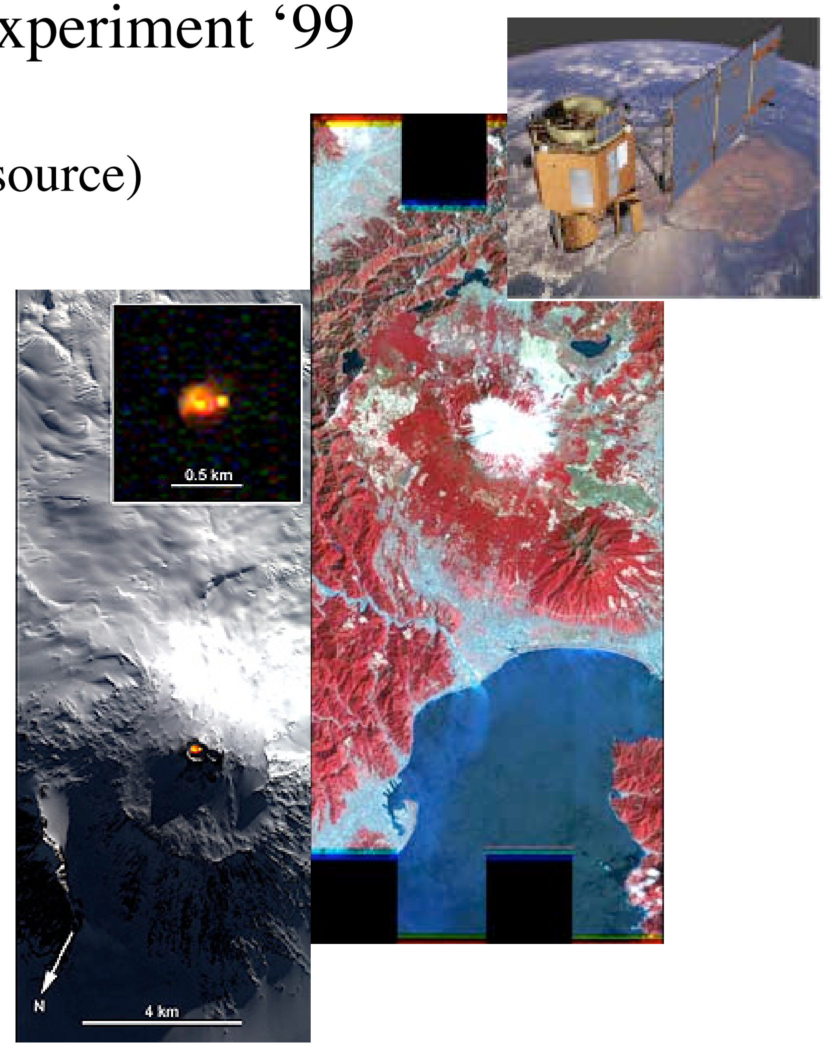




NASA Applications

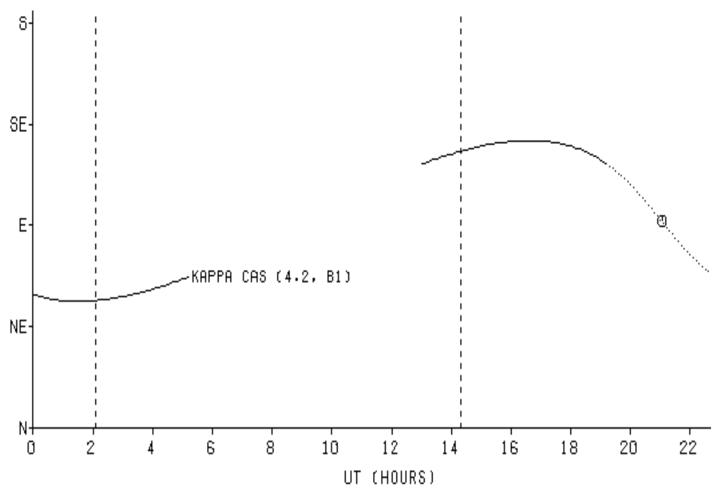
- Constraint Reasoning used in Missions
  - DS1: RAX – Remote Agent Experiment '99
    - Spacecraft pointing constraints
    - Onboard memory (renewable resource)
    - Thrust accumulation constraints
  - EO-1 ScienceCraft '04
    - Thermal duty cycle constraint





NASA Applications

- Constraint Reasoning used in Missions
  - Automated telescope scheduling (ATIS) '99
    - sin h = sin □ sin □ + cos □ cos □ cos (□-L-□)
  - Hubble Space telescope scheduling '94
    - Orbit period constraints
    - Sun and Earth occlusion constraints



Ames Research Center

NASA Applications

- Mission-oriented research
  - Earth-observing satellite scheduling project (EOS)
  - SOFIA flight scheduling project (SOFIA)
  - Contingent Planning for Mars rover operations
  - Personal Satellite Assistant (PSA)
  - Spoken Interface Prototype for PSA
  - Space Interferometry testbed (SIM)
  - Unmanned Helicopter Surveillance Scheduling
- Mission-directed Research
  - UAV Autonomy Architecture
  - Intelligent Deployable Execution Agent (IDEA)
  - LORAX Rover Power budgeting
  - Image processing planning (ImageBot)

Ames Research Center

NASA Applications

- Some common themes
  - Heterogeneous constraints and optimization
    - Mixes of discrete, continuous
    - Small -arity and large -arity
  - Complex constraints
    - DEs, tightly coupled constraints (e.g. resources)
    - Qualitative and quantitative uncertainty
  - Dynamic constraints
    - Added and retracted all the time!
  - Integrated constraint solvers
    - Solvers in Planners, schedulers
    - Both ground systems, on-board systems
    - Distributed solvers

Ames Research Center

NASA Technology

- Constraint-based Planning
  - Generalization of classical AI planning
  - Heavy use of constraint based modeling and constraint reasoning
  - PLASMA: Plan State Management Architecture
    - Ground tools and onboard systems
- Modeling issues
- Constraint propagation
- Temporal Flexibility and Resources

Constraint-Based Planning

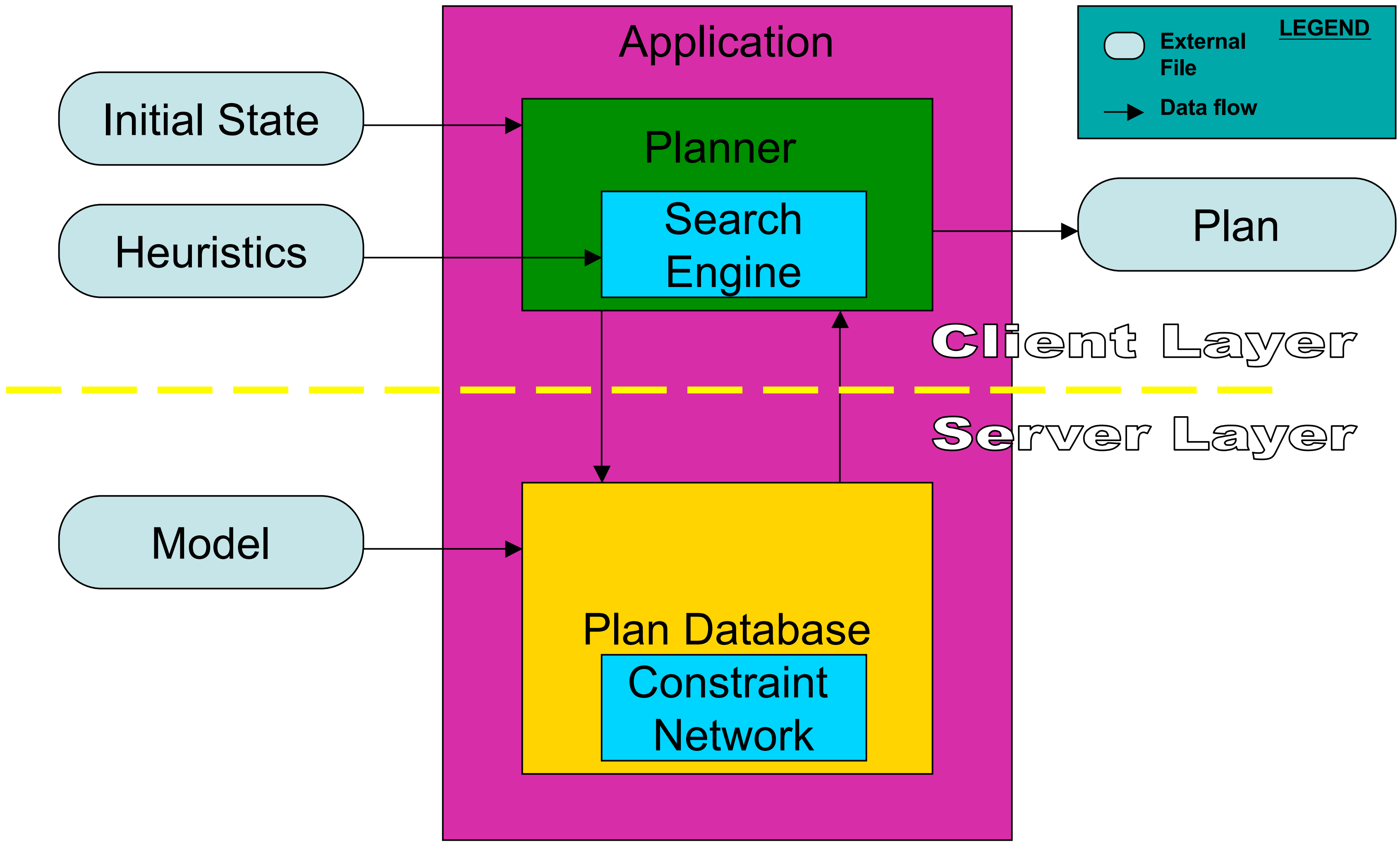
- A ***Domain Model***:
- defines parts of the plan
- defines necessary relationships among them for valid plans

- The ***Plan Database*** :
- maintains current plan
- maintains mapping between plan and constraint network
- supports plan modification and constraint inference

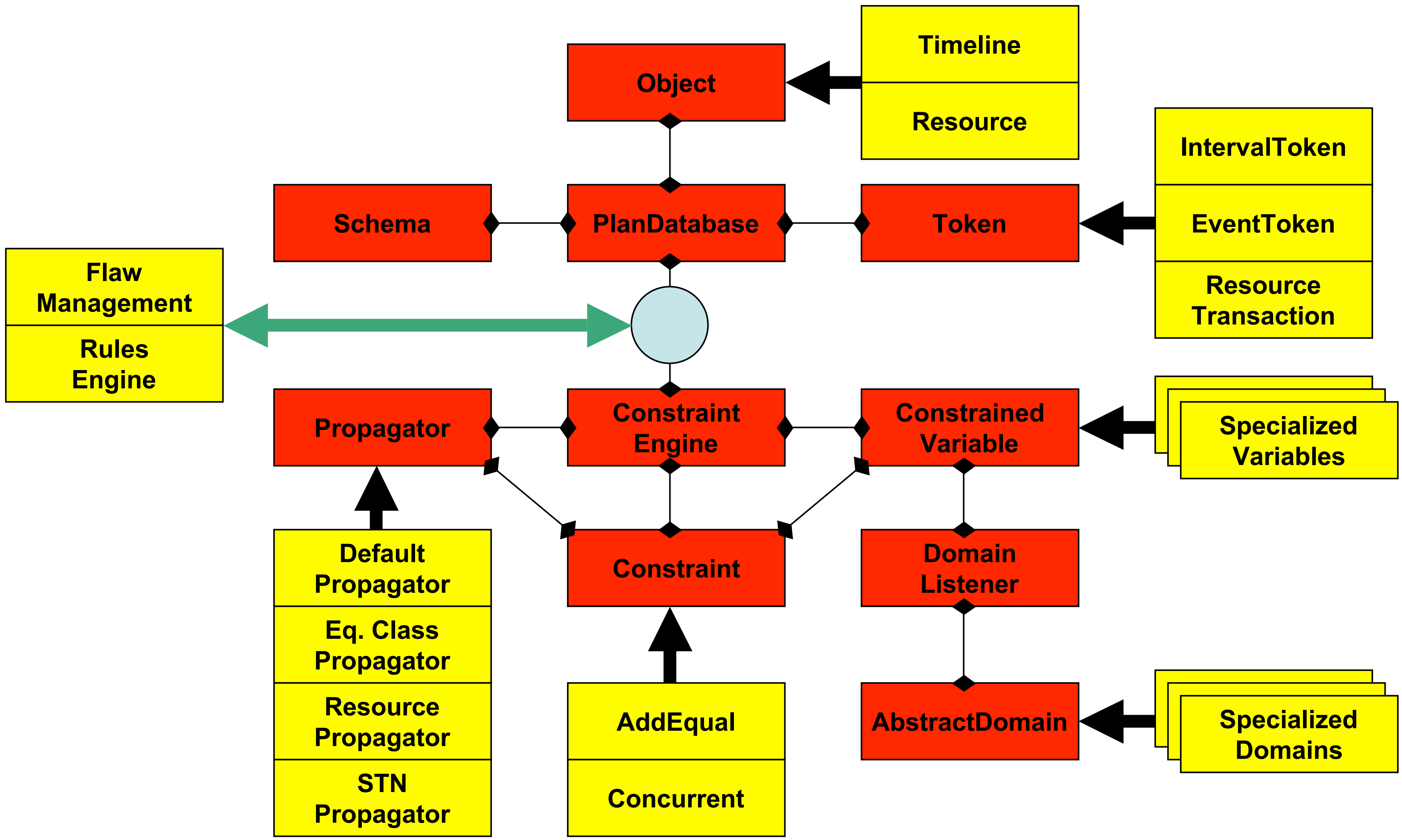
- The ***Planner***:
- checks status of current plan
- decides how to modify the plan

Ames Research Center

Constraint-Based Planning Application Architecture



PLASMA
Framework & Components



Modeling Paradigm

- Class: general object description
  - Object: class instance
  - Predicate: state an object can be in
  - Rules: relationships between objects
- Variables
  - Predicates represented by variables
    - Start, end (timepoints), duration
    - Parameters of predicates
- Rules are templates for constraints on variables
  - Sequencing of states on same object imposes constraints
  - Appearance of state in plan constrains other states

Ames Research Center

Sample Model Fragment

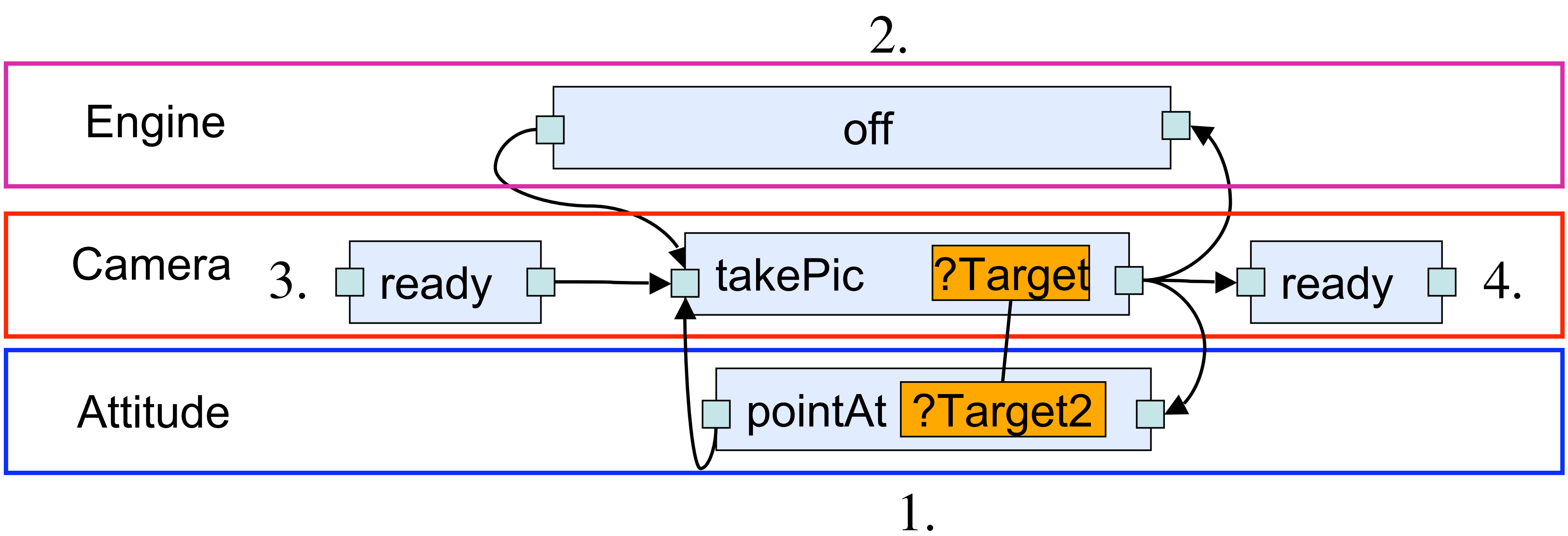
- Camera::TakePic{
- // 1.Attitude must be constant throughout
- contained_by(Attitude.pointAt at);
- eq(at.location, rock);

- // 2. Engine must be off throughout
- contained_by(Engine.off o);

- // 3. Preceded by readying operation
- met_by(Ready r);

- // 4. Succeded by stowing the instrument
- meets(Stow c);
- }

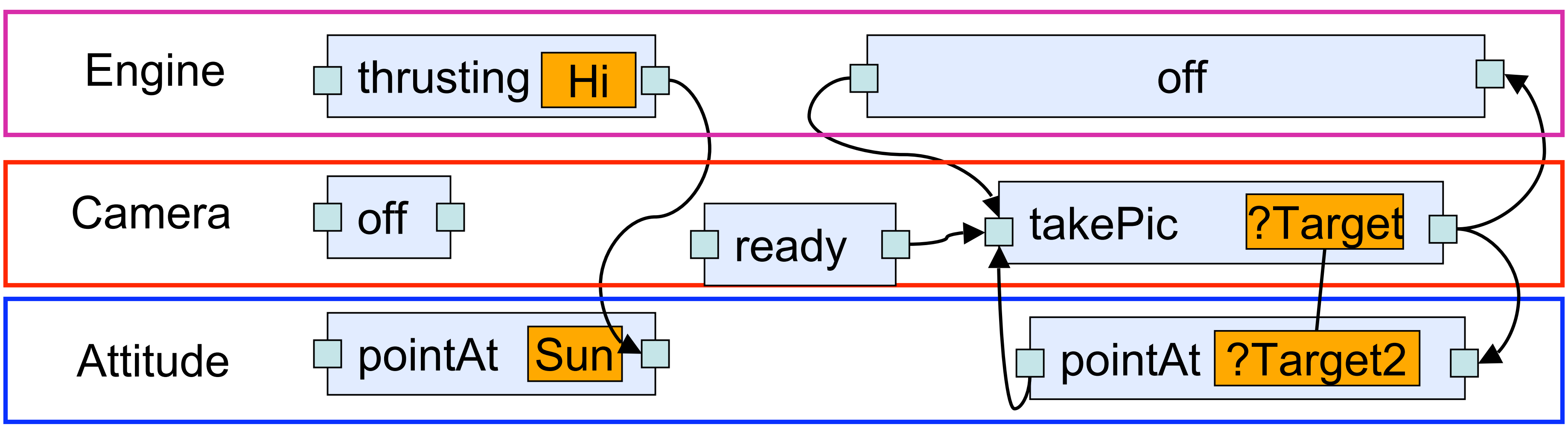
Sample Model Fragment



Ames Research Center

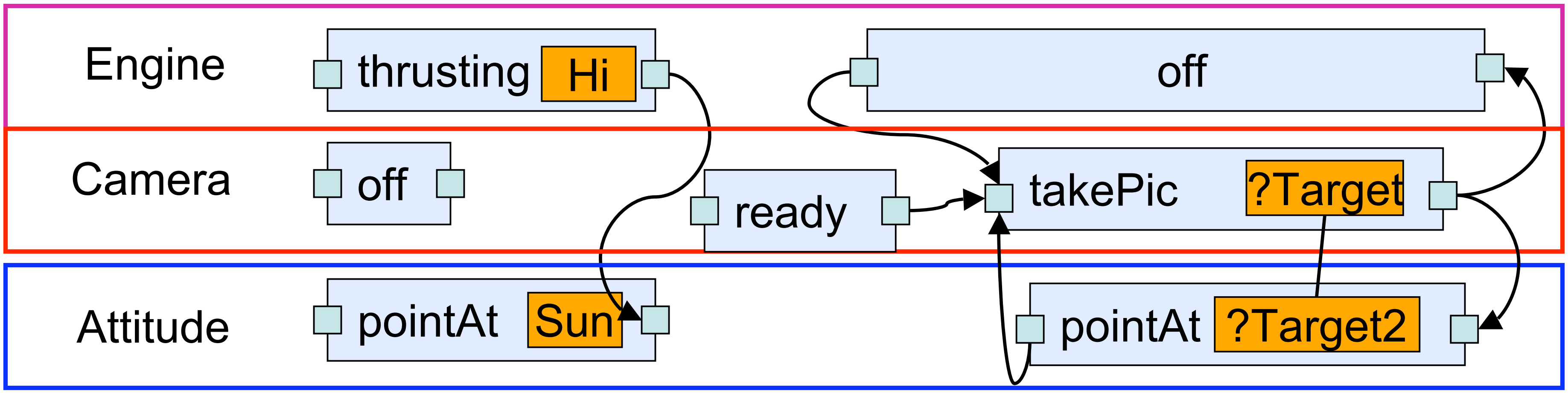
Plan Representation

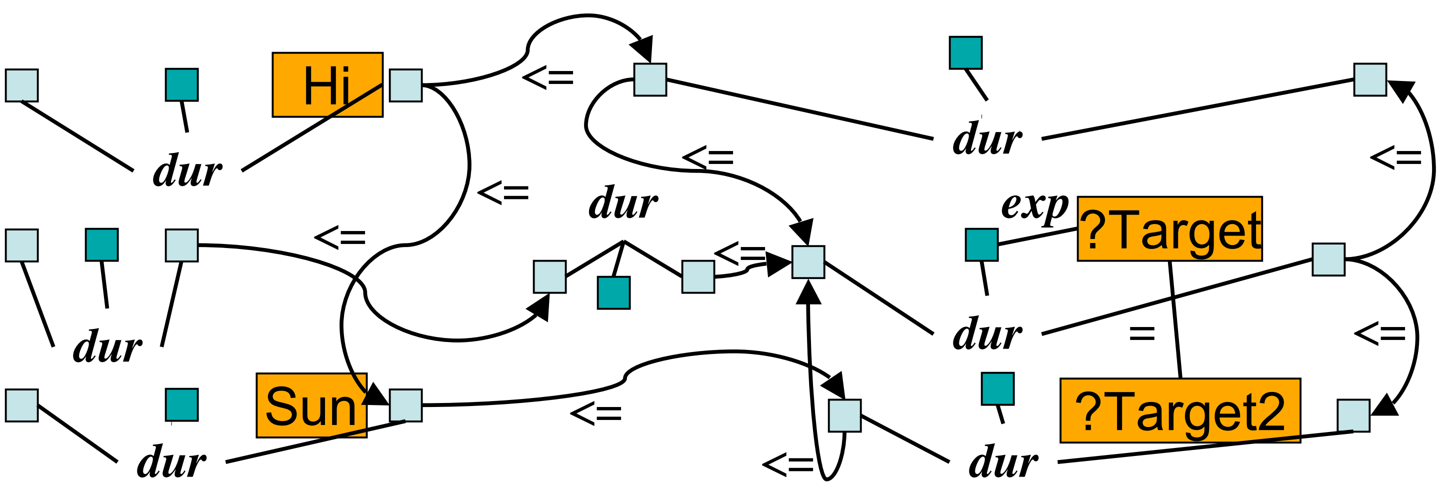
- ***Timelines*** are class instances, and enforce temporal mutual exclusion over an object's state
- ***Parameterized Predicates*** describe actions and states
- ***Time Intervals*** have Start, End and Duration
- ***Token*** is a Parameterized Predicate over a Time Interval
- ***Constraints*** defined between Time Points, Parameters



Plan Representation

- Every partial plan is mapped to a CSP





Ames Research Center

The Planning Process: Flaw/Decision Model

Variable Decisions (resolve unbound variables):

- Specify (var, val) / Reset (var)

Token Decisions (resolve inactive tokens):

- Activate(Token t) / Deactivate(Token t)
- Merge (Token t1, Token t2) / Split(Token t1)
- Reject(Token t1) / Reinstate (Token t1)

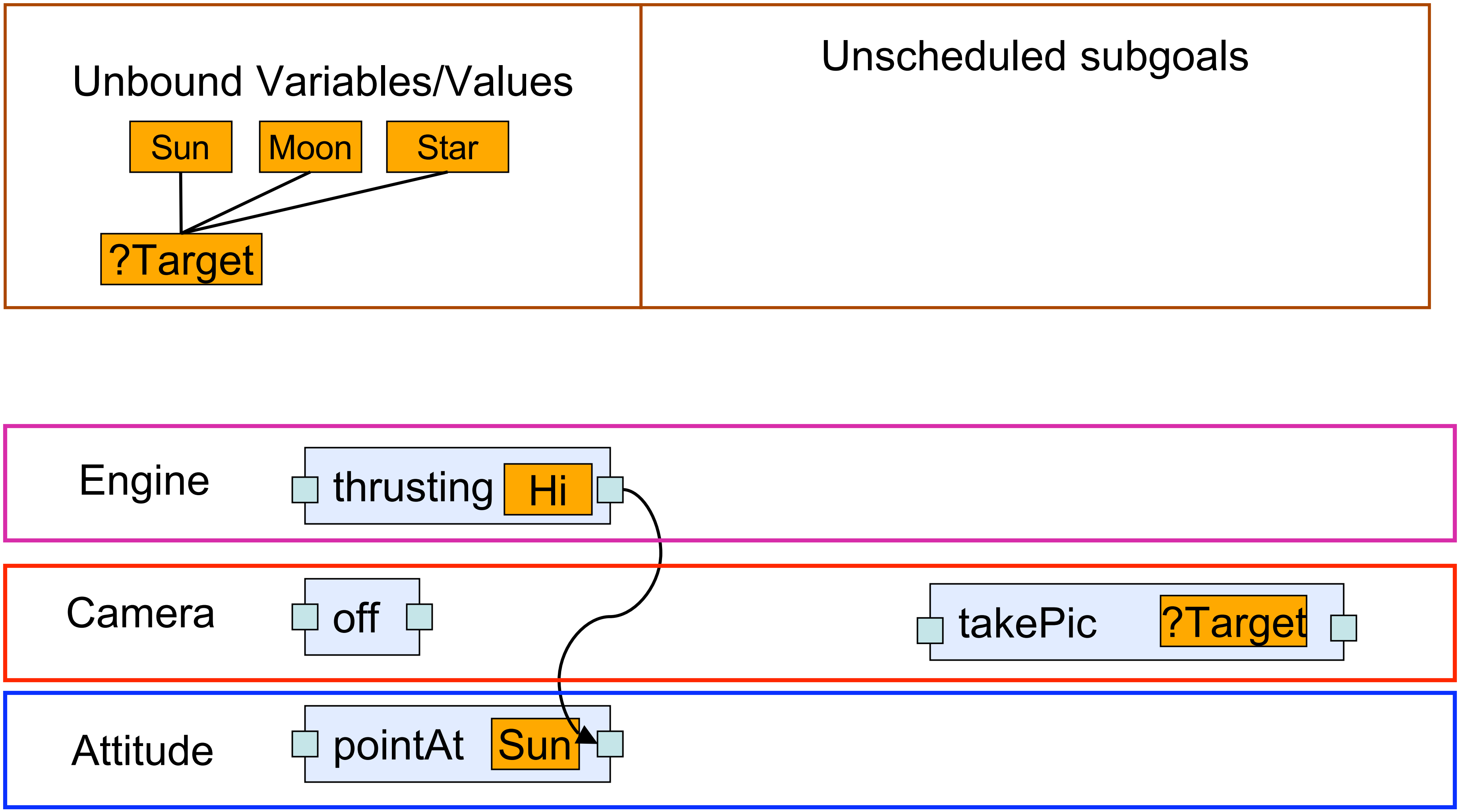
Object Decisions (resolve when Object hasTokensToOrder):

- Constrain(Object o, Token t) / Free(Token t)
- Constrain(Object o, Token t1, Token t2) / Free(Token t1)

The Planning Process

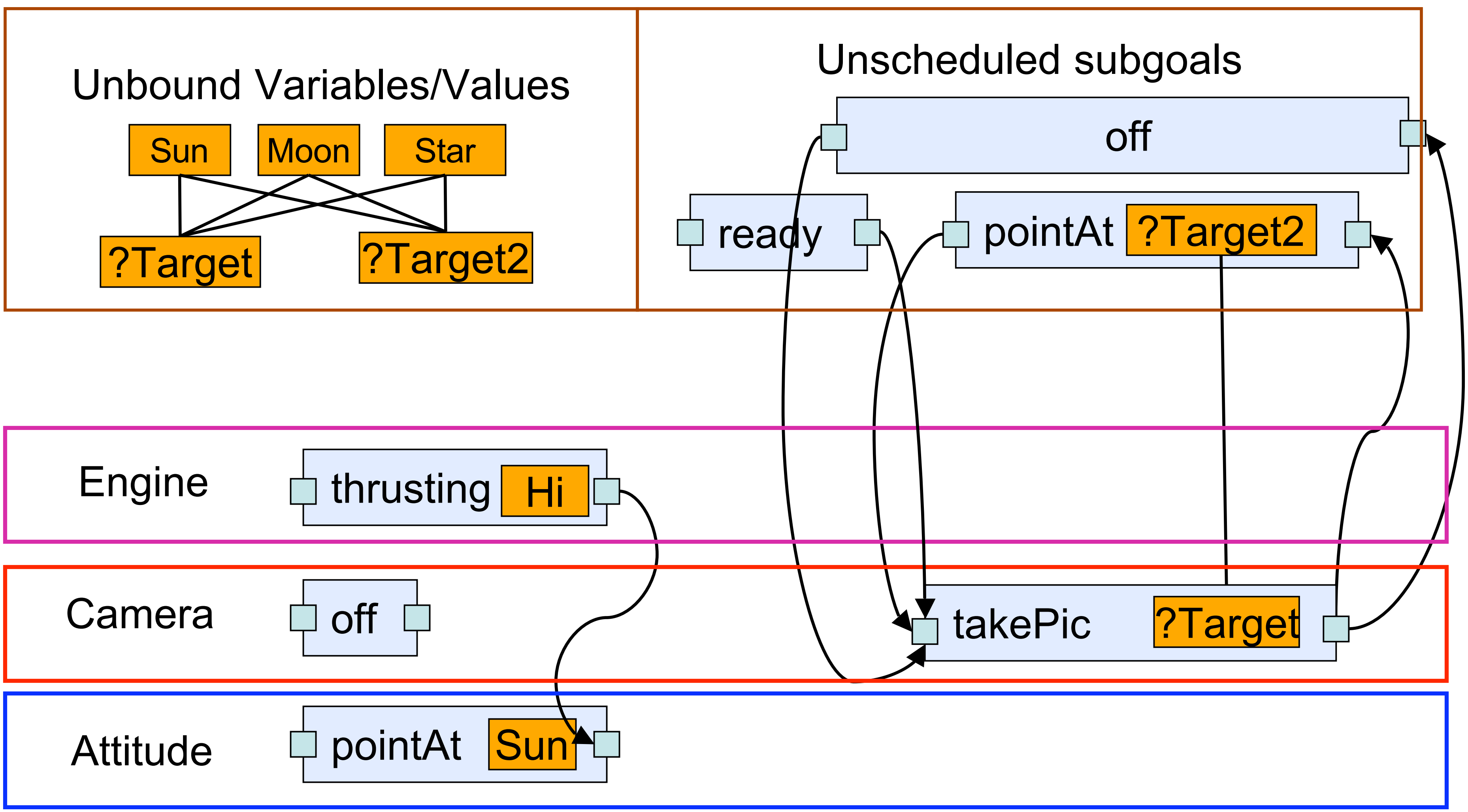
- All Flaws/Decisions can be viewed as CSP variable assignment options
  - Token decisions: merge + rejection straightforward, sequencing requires enumeration of options
  - Object assignment options straightforward
- As partial plan evolves, CSP changes according to rules
  - Thus, Planning is equivalent to solving a DCSP
  - Unlike "classical" DCSP (Mittal & Falkenhainer 1990)
    - Allowed to create new CSP variables, modify domains of existing variables
    - Have rules describing CSP modifications to consult during solving

Insert takePic



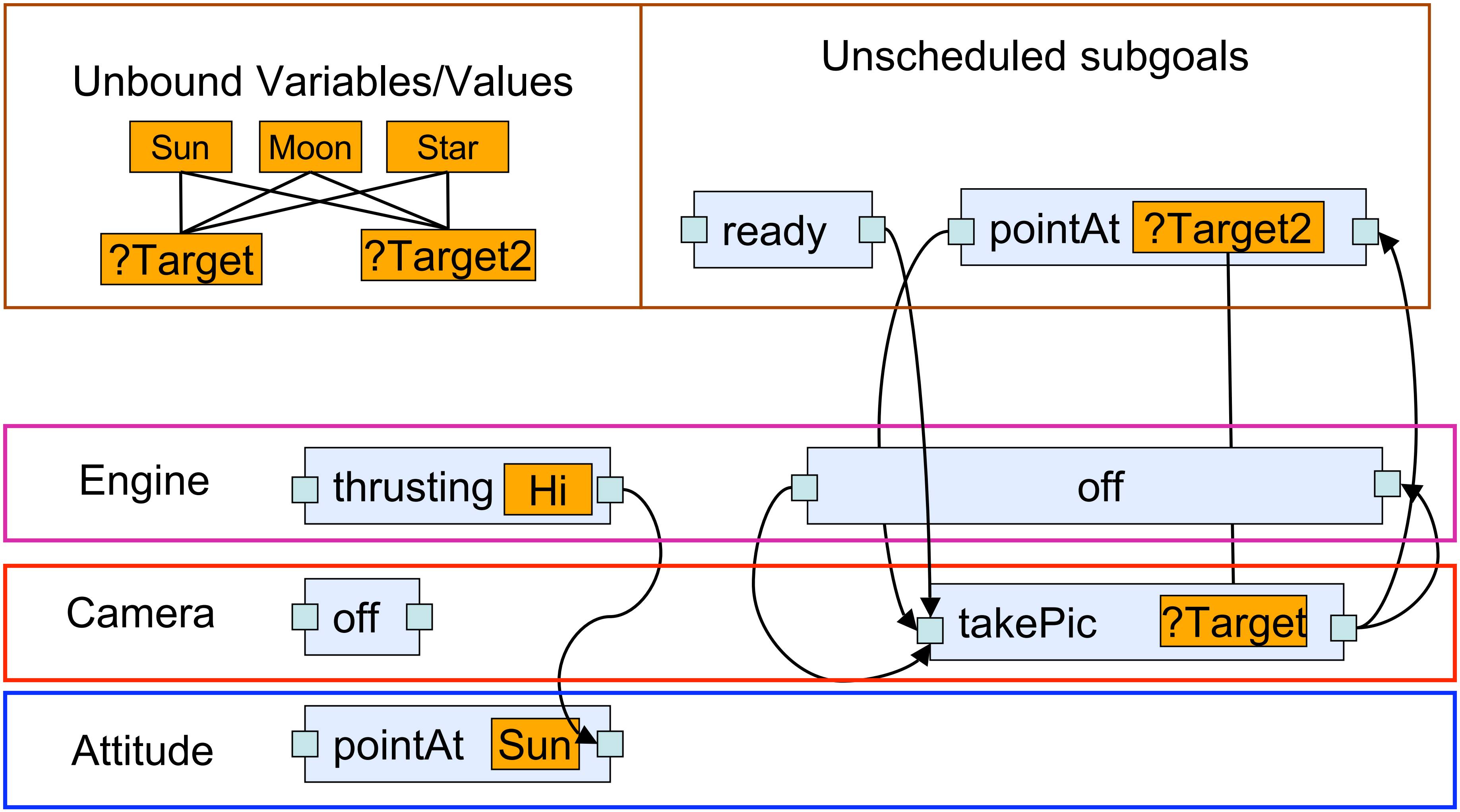
Ames Research Center

Expand takePic subgoals



Ames Research Center

Insert off



Ames Research Center

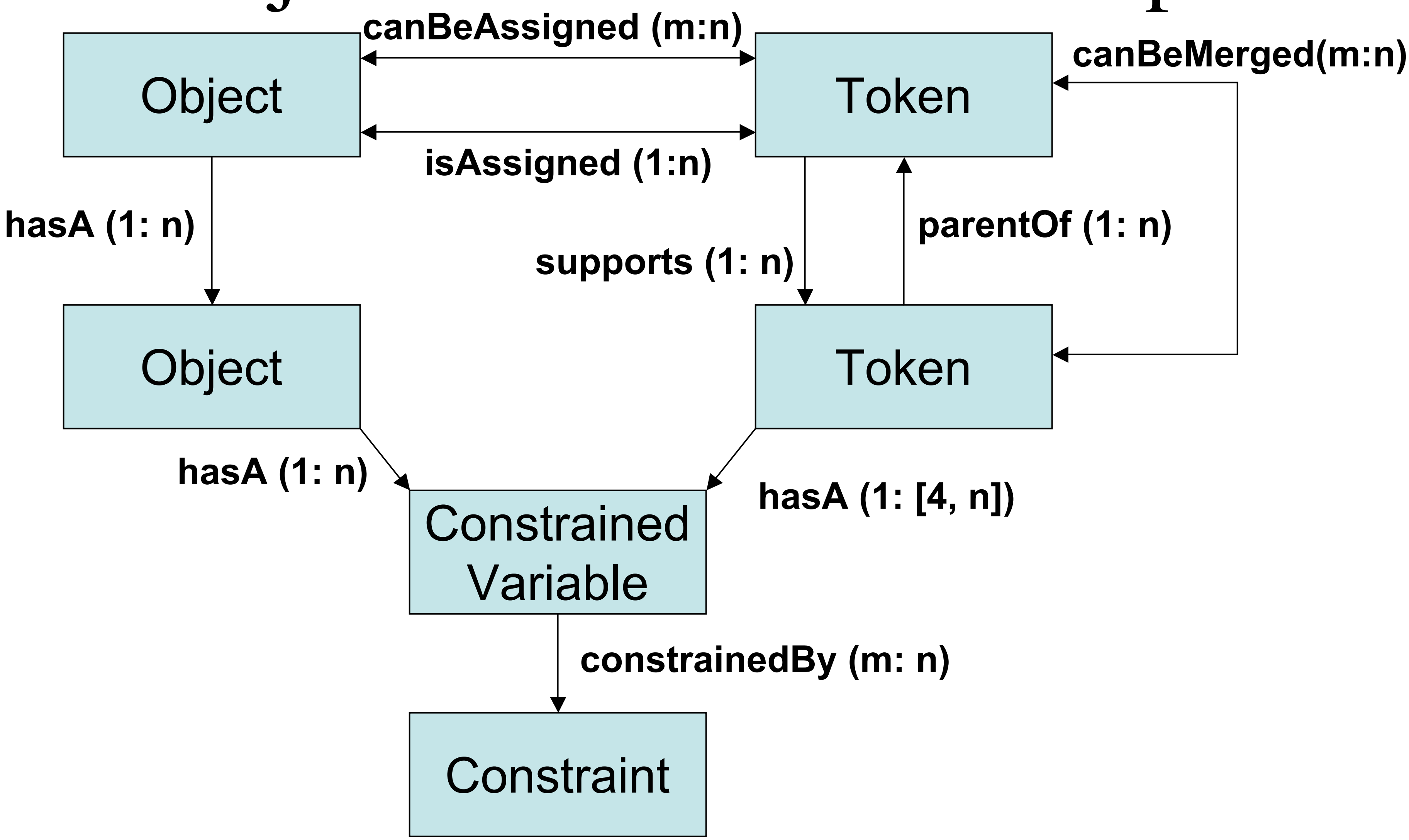
Objects with and without Tokens

Object	Member Variables (Static w.r.t. Time)
Rock	name(rock4)
	x(3)
	y(9)

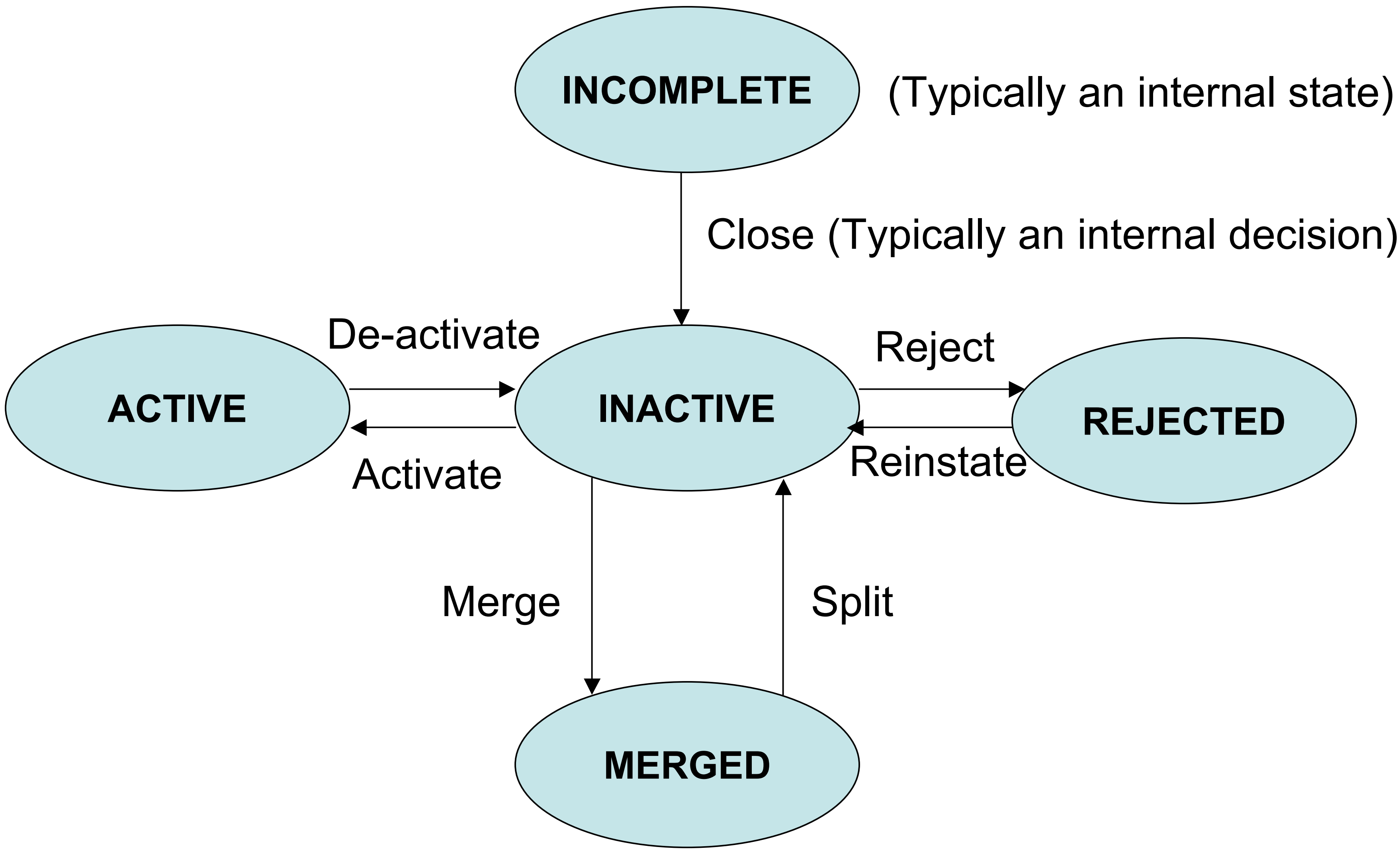
Object	Member Variables (Variable w.r.t. Time)		
Navigator	At(lander)	Going(lander, rock4)	At(rock4)

Object	Member Variables (Variable w.r.t. Time)			
Instrument	Stowed	Unstow	Place(rock4)	TakeSample(rock4)

Object - Token Relationships



Token State Transition Model



Implications

- Modeling & Constraint Architecture
  - How to do the CSP representation?
- Heterogeneous constraint propagation
  - Scheduling consistency enforcement
  - "Heterogeneous" consistency
    - What if half my CSP is AC and the other half BC?
- Heuristics
  - Do binary CSP heuristics apply?
  - Do static CSP heuristics apply?
- Hardness of problems
  - Does phase transition work apply?

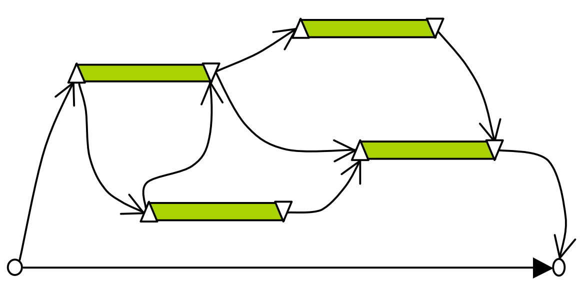
Modeling Time (1)

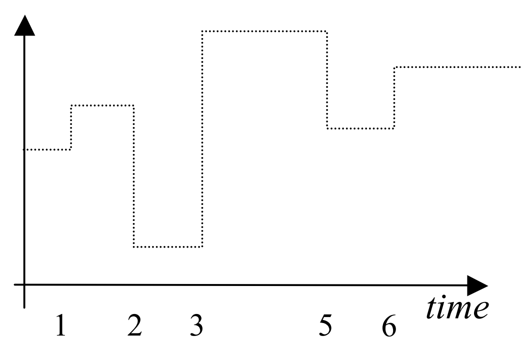
- Temporal constraints or systems of linear constraints?
  - Feasibility of linear constraints: Gaussian elimination
  - A general mechanism for all such constraints
  - ...but STNs more efficiently handled with Shortest path algorithms
  - Requires specialized propagation to general linear constraint solver

Ames Research Center

Modeling Time (2)

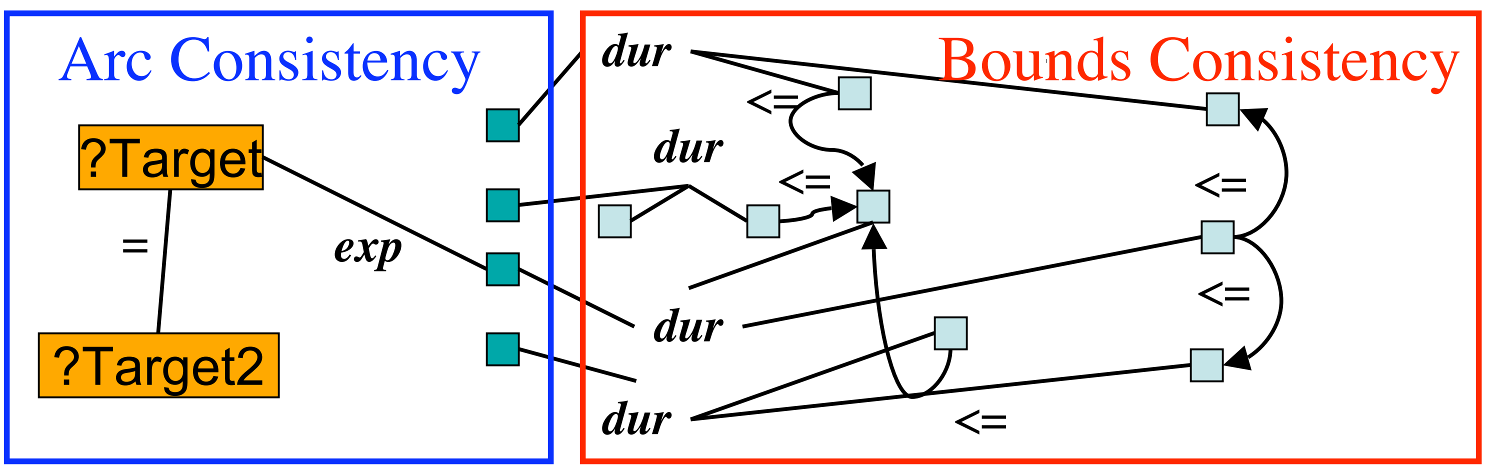
- Temporal and Resource Constraints
  - Laborie 2001, Muscettola 2002 & 2004, Frank 2004
    - Constraint checks and consistency enforcement
    - Temporal and resource constraints
  - Works for scheduling problems
  - ...but problematic for planning problems
    - Actions that impact one of a set of resources
    - Possible unification of actions
    - As-yet ungenerated actions
  - Polynolmial time, but expensive





Modeling Time (3)

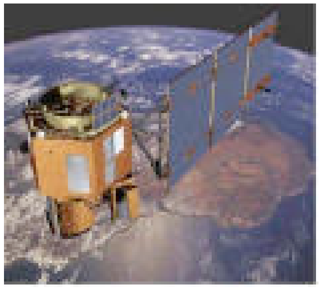
- That pesky duration constraint
  - A "hybrid" consistency representation
    - STNs enforce Bounds consistency
    - ...but other constraints act on duration...
    - ...requiring mapping between bounds-consistent start & end, arc-consistent duration variable
  - ...or a more complex model?
    - Epillitis (Tsamardinos et al. 2003) directly handles disjunctive STNs (DTNs)
    - ...but some algorithm tailoring required for efficiency



Ames Research Center

Modeling Time (4)

- Satellite scheduling
  - Consider satellite scheduling with
    - known orbit
    - discrete observation choices
    - pointable instrument

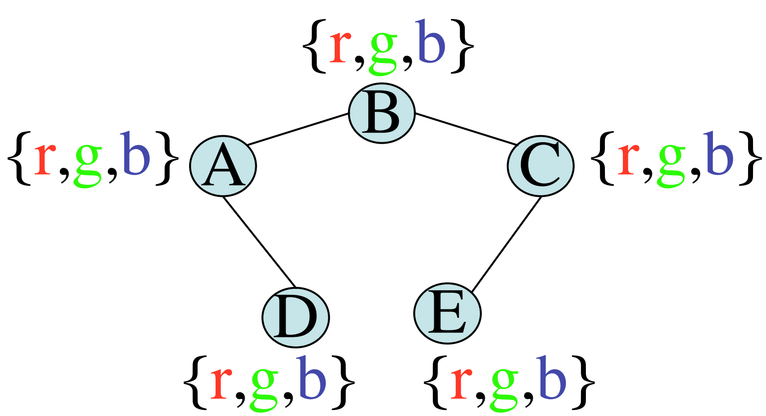


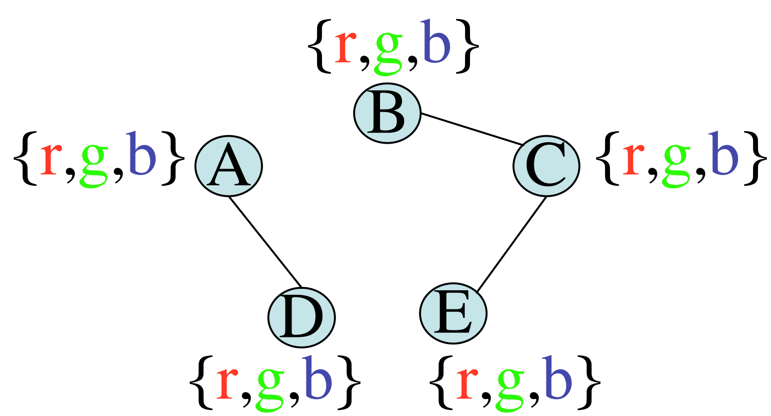
  - No reason to constantly solve trigonometric constraints!
    - Can pre-compile feasible slews
      - If you do x can't slew in time to do y
    - Treat this constraint as a binary CSP
  - We did this wrong the first time
    - We failed to learn from Verfaillie et al.
    - ...and we paid!

Ames Research Center

Consistency Matters

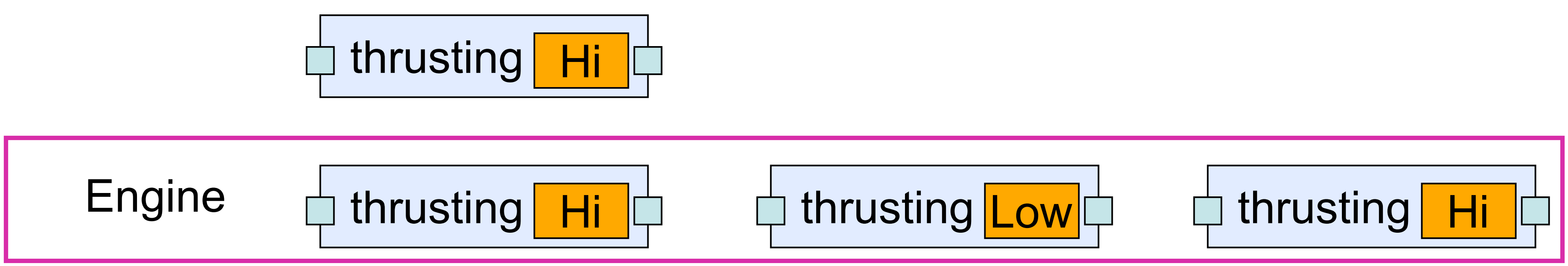
- Equivalence classes and GAC
  - Faster to propagate equivalence class than lots of "connected" binary equalities
  - ...but now DFS required to maintain connected equalities
  - ...and since many equalities hold on timepoints...
- The same holds for AllDiff
  - ...but need to maintain cliques of AllDiff variables...
  - ...and where's AllDiff in our models?





Matters of State

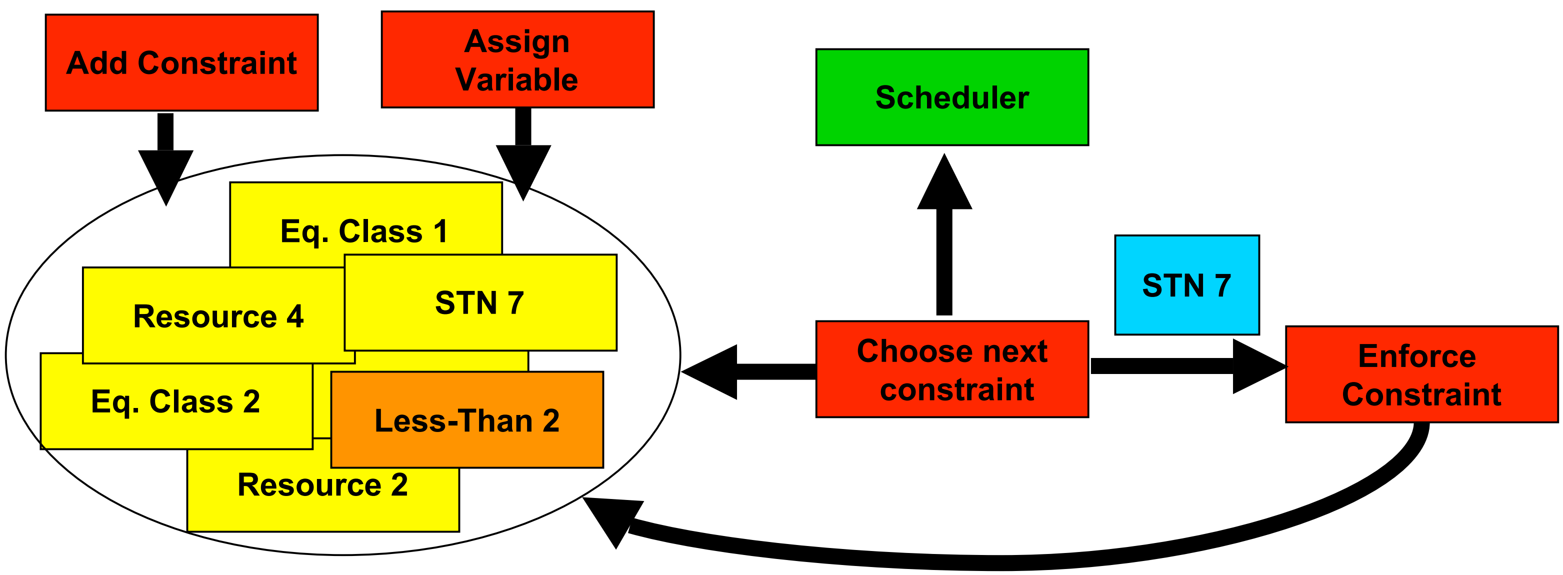
- Representing Token Insertion Decisions (Frank et al. 2000)
  - Timelines enforce mutex; Where can a token go?
  - Dynamic variable domains
    - Can’t store “domain” of a token
    - What does this do to nonchron. search and nogood reasoning?
  - Forward checking rather than AC to generate slots on-the-fly
- Other representations have other problems



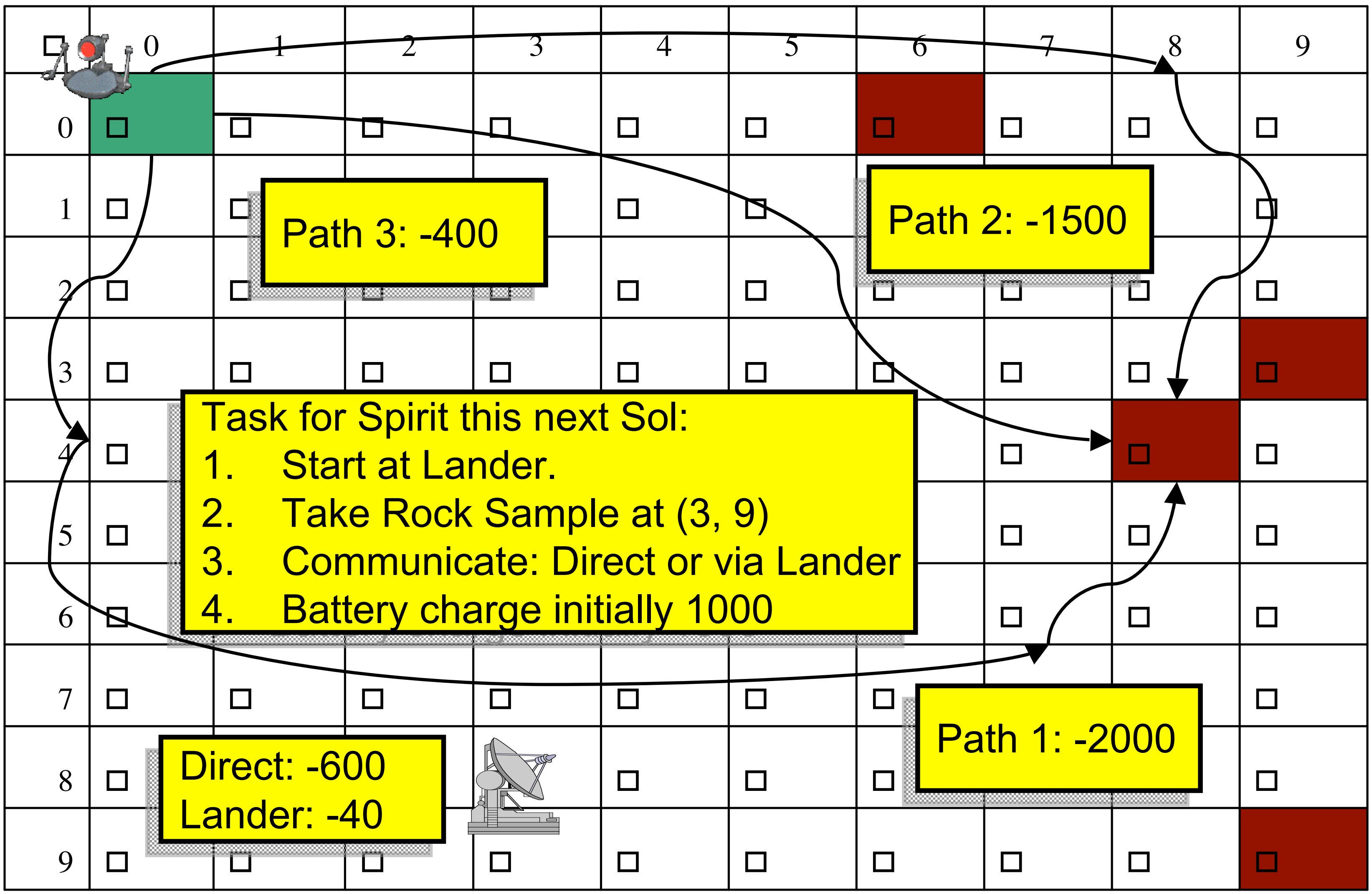
Ames Research Center

Handling Heterogeneous Constraints

- Distinguished constraint classes
  - Generic, Temporal, Equivalence classes, Resources
- Rules engine
  - Adds and removes constraints
- Triggering propagation via events
  - Scheduling of execution can be defined by user



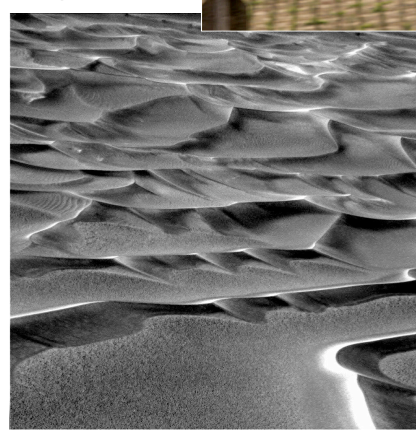
PLASMA Demo: Rover Rock Sampling



Ames Research Center

Some New Frontiers

- Scheduling propagation
- Optimizastion
- Heuristics
- New “Hybrids”
- Limits of the application



...SO WHAT?

- Constraints MATTER.
  - Used in “real” systems.
  - Solves “real” problems.
- ...but the context MATTERS too.
  - Space isn’t manufacturing isn’t academia.
  - Robotics isn’t biology isn’t telecom.
  - Remember your customer.

Ames Research Center

Bibliography

- J. Frank and A. Jonsson. Constraint-Based Attribute and Interval Planning. J. Constraints 8(4) 2003
- J. Frank. Bounding the Resource Availability of Partially Ordered Events with Constant Resource Impact. CP 2004
- A. Bachmann et al. PLASMA: A Constraint-Based Planning Architecture. (Demonstration) CP 2004.
- N. Muscettola. Computing the Envelope for Stepwise-Constant Resource Allocations. CP 2002
- N. Muscettola. Incremental Maximum Flows for Fast Envelope Calculation. ICAPS 2004
- P. Laborie. Algorithms for propagating resource constraints in AI planning and scheduling: Existing Approaches and New Results. Artificial Intelligence 143, 2003.
- Y, Tsamardinos and M. Pollack. Efficient Solution Techniques for Disjunctive Temporal Reasoning Problems. Artificial Intelligence, 151(1) 2003
- J. Frank et al. On Reformulating Planning as Dynamic Constraint Satisfaction. (Extended Abstract) SARA 2000
- A. Jonsson and J. Frank. A Framework For Dynamic Constraint Reasoning using Procedural Constraints. ECAI 2000
- S. Mittal and B. Falkenhainer. Dynamic Constraint Satisfaction Problems. AAAI 1990
- E. Bensasa and M. Lemaitre and G. Verfaillie. Earth Observing Satellite Management. J. Constraints 4(3) 1999