Goal-based Operations

Michel D. Ingham, Sc.D.

Jet Propulsion Laboratory
California Institute of Technology

Ground System Architecture Workshop 2006

"Toward a Standard for Goal-Based Operations" Working Group

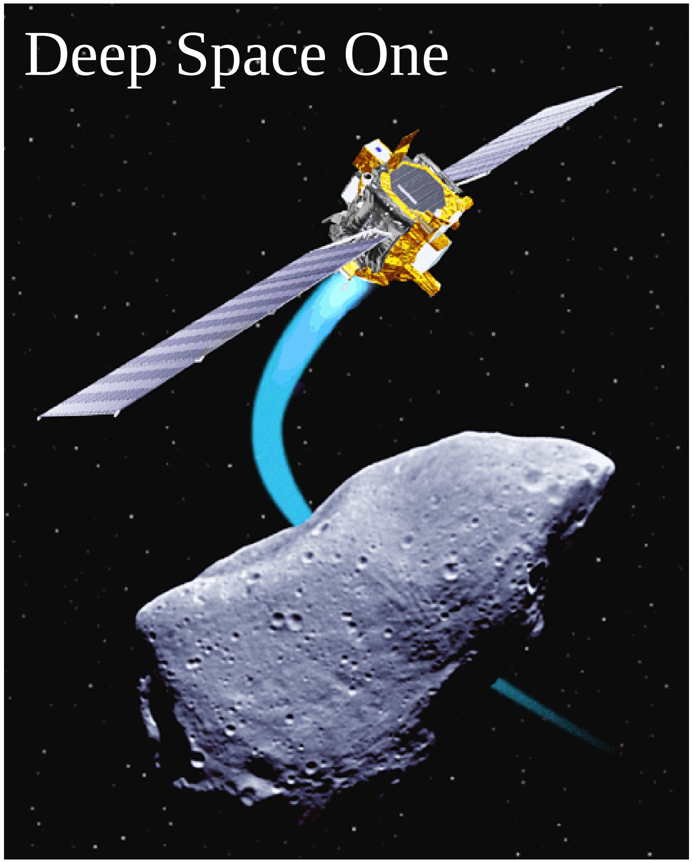
Manhattan Beach, CA

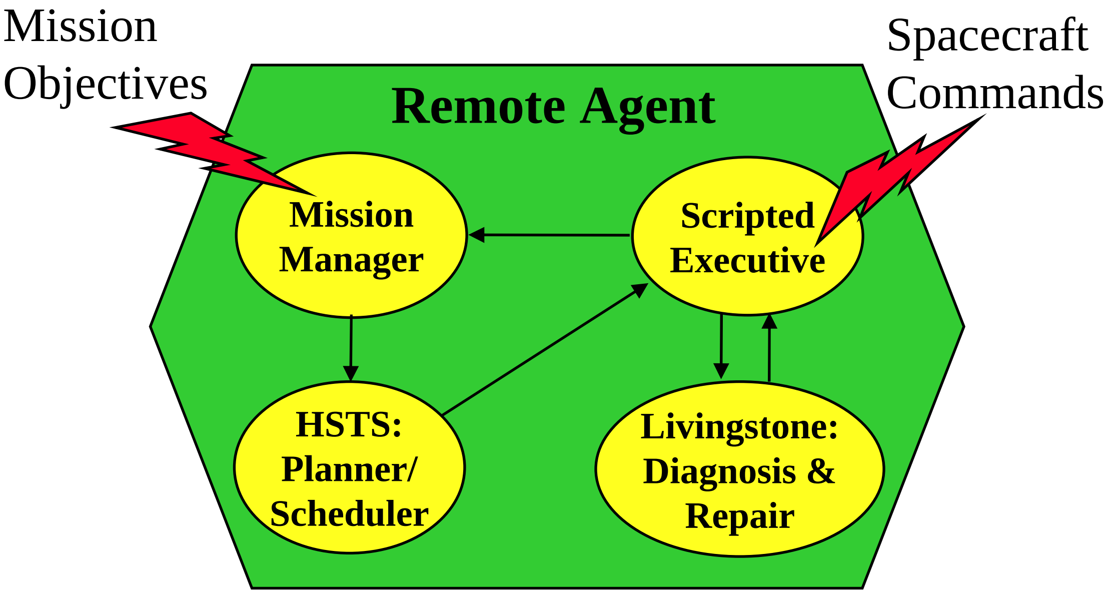
March 29, 2006

Objectives

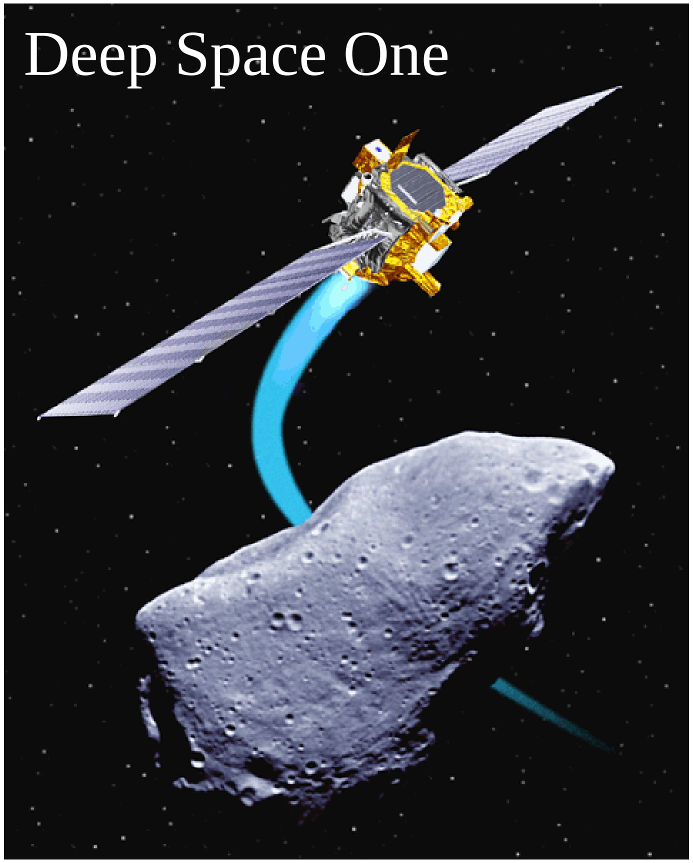
- Describe a few NASA projects that have taken first steps in the area of goal-based operations
- Highlight the benefits that these initiatives have demonstrated
- Lay out some Challenge Questions that probably need to be answered for this approach to really take hold

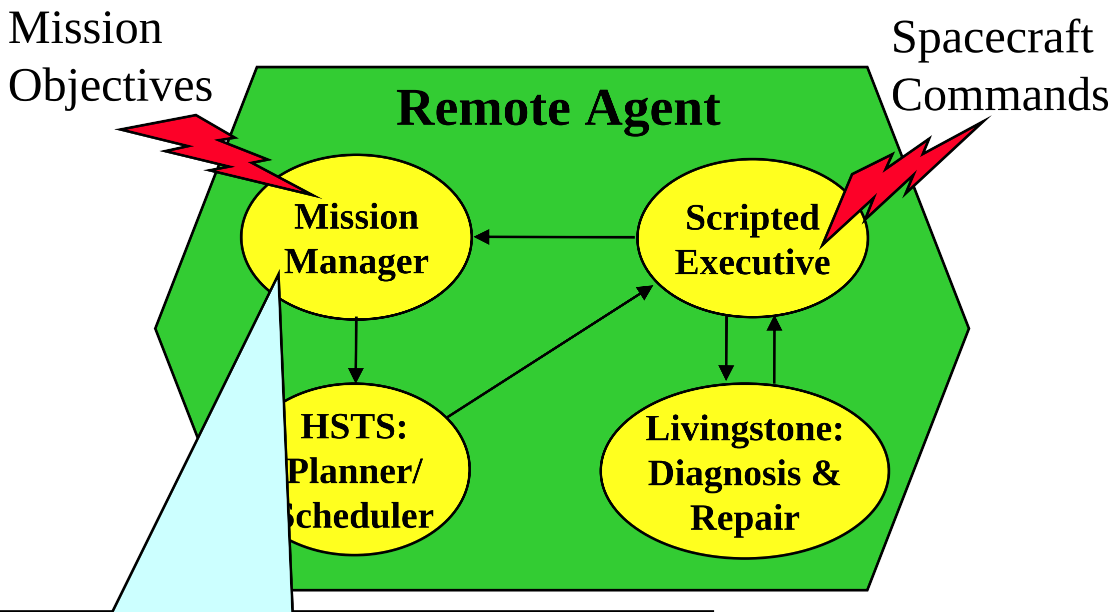
Steps in the Right Direction (1)





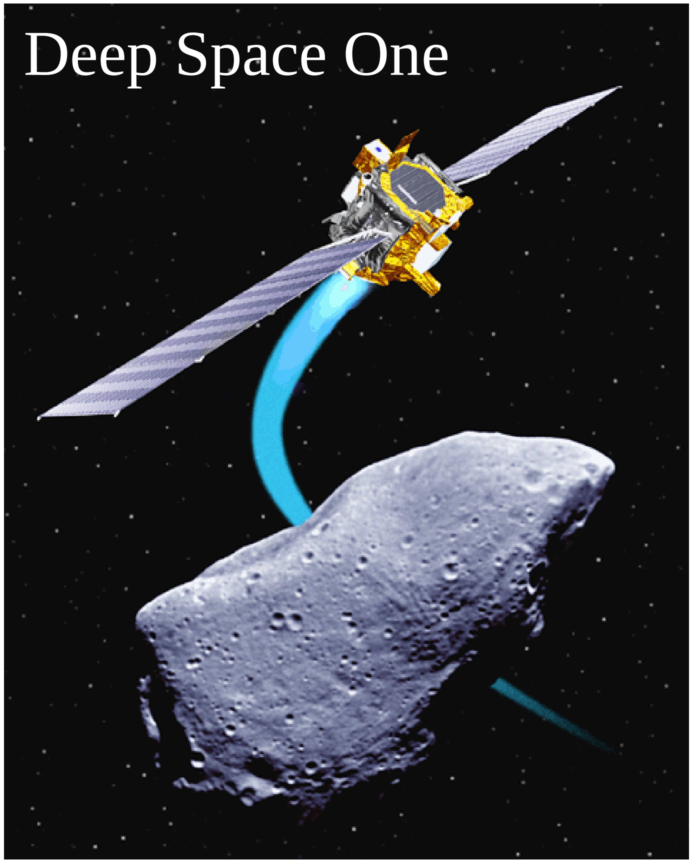
Steps in the Right Direction (1)

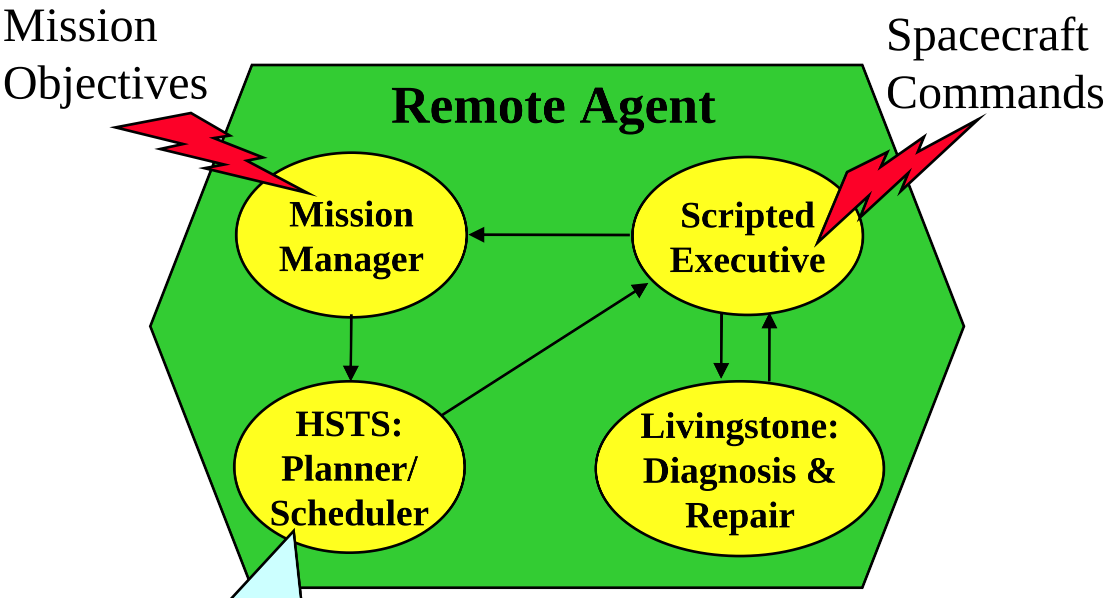




Sends Mission Objectives as *high-level goals* to the Planner/Scheduler.

Steps in the Right Direction (1)





Mission Objectives

Spacecraft Commands

Remote Agent

Mission Manager

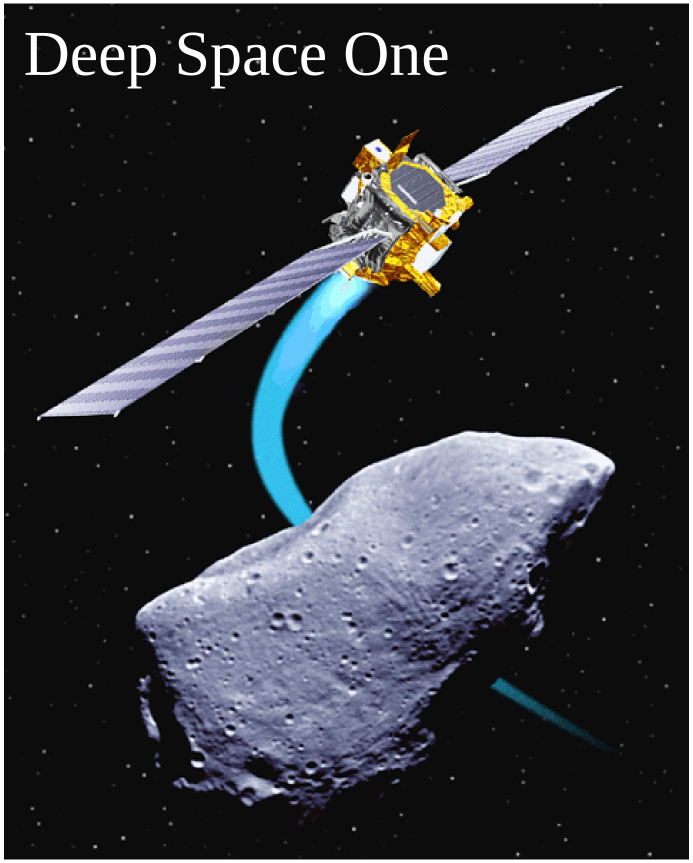
Scripted Executive

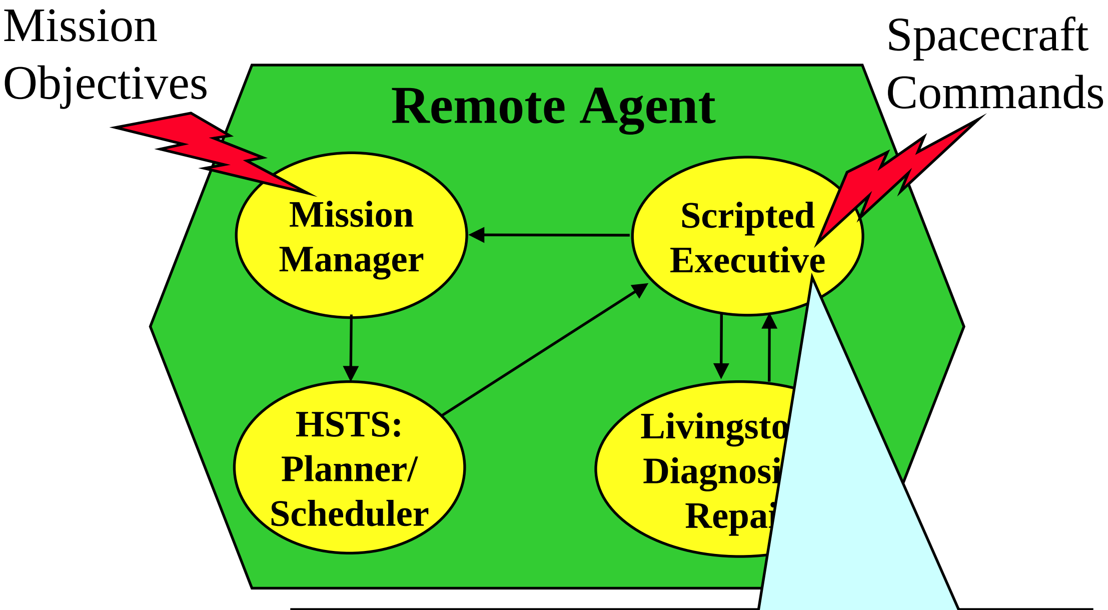
HSTS: Planner/ Scheduler

Livingstone: Diagnosis & Repair

Plans and schedules detailed tasks (lower-level goals) to achieve the high-level goals.

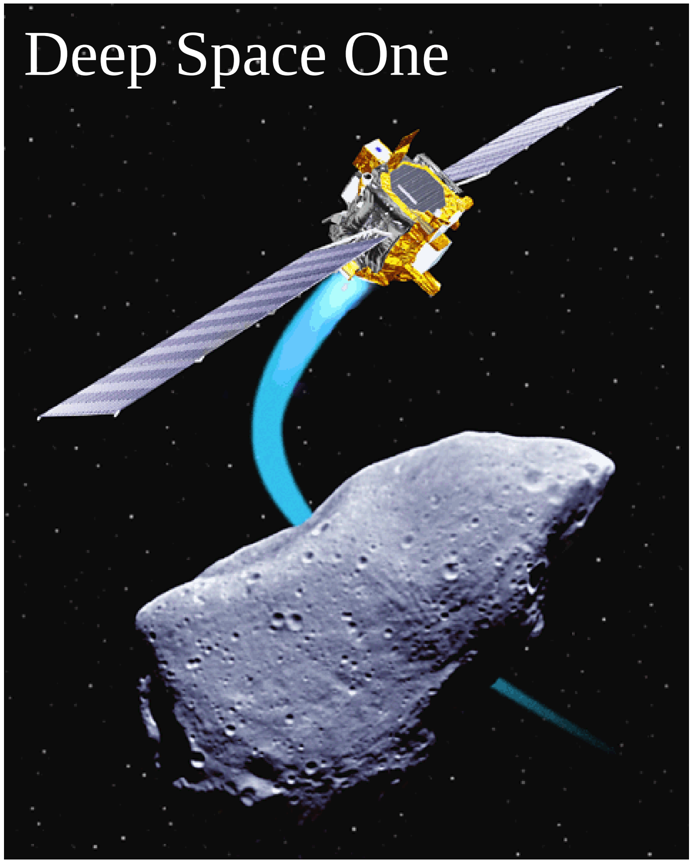
Steps in the Right Direction (1)


Deep Space One

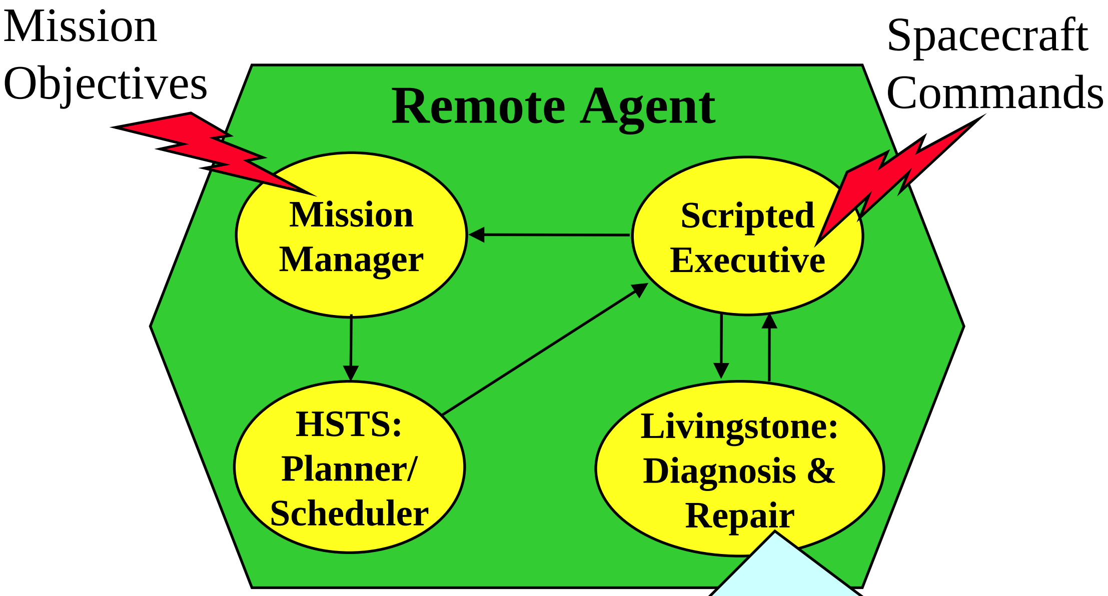


Executes scripts associated with lower-level goals, issues appropriate commands.

Steps in the Right Direction (1)

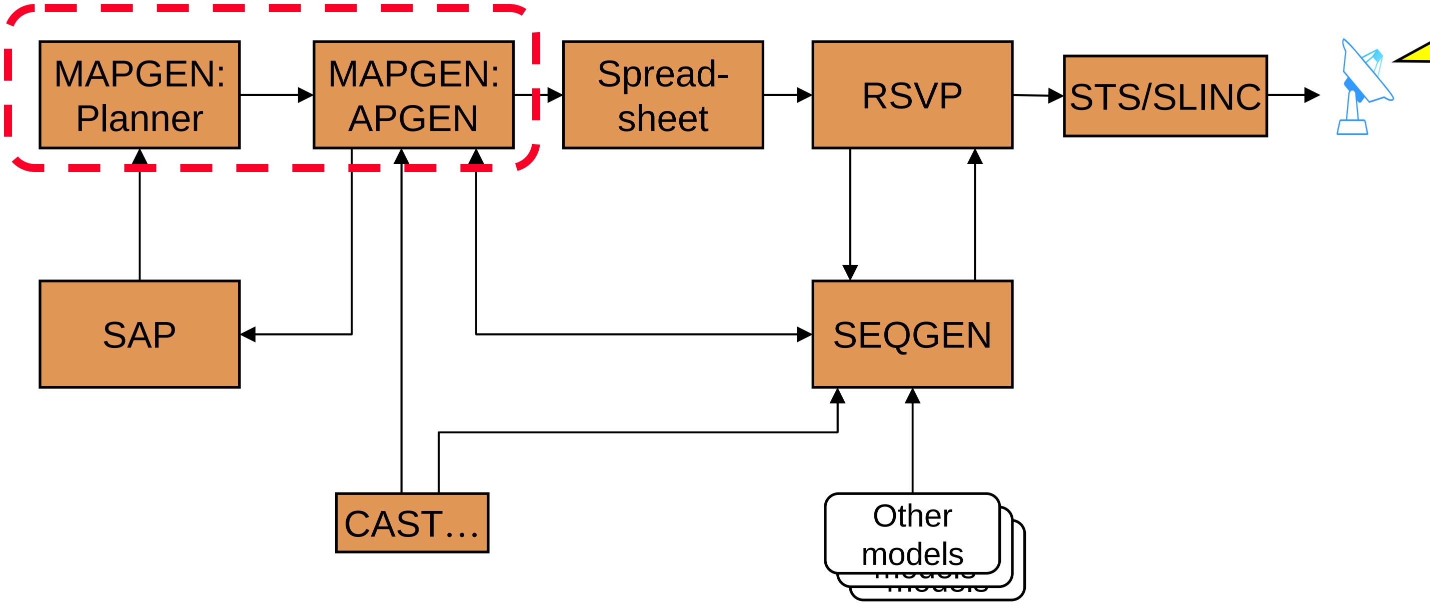


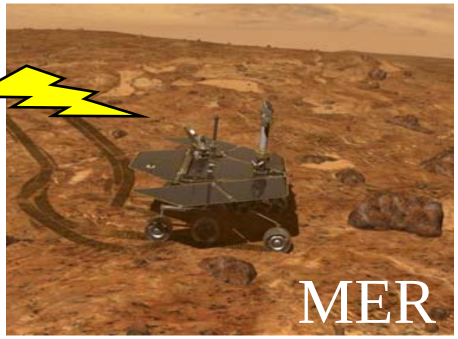
Deep Space One



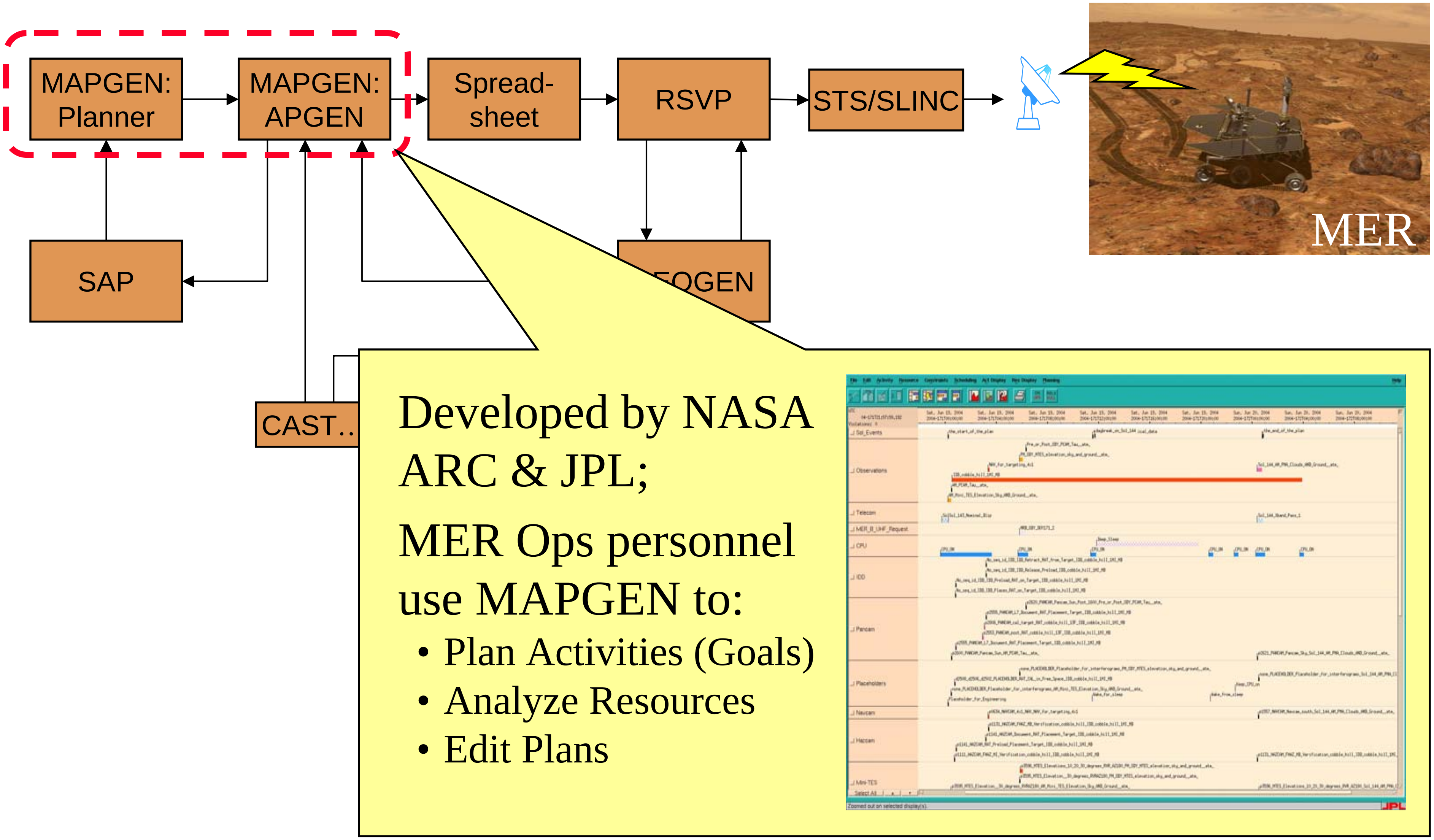
Provides state estimates and suggests reconfiguration commands to Exec, by reasoning through a declarative model of the spacecraft system.

Steps in the Right Direction (2)





Steps in the Right Direction (2)



Developed by NASA ARC & JPL;

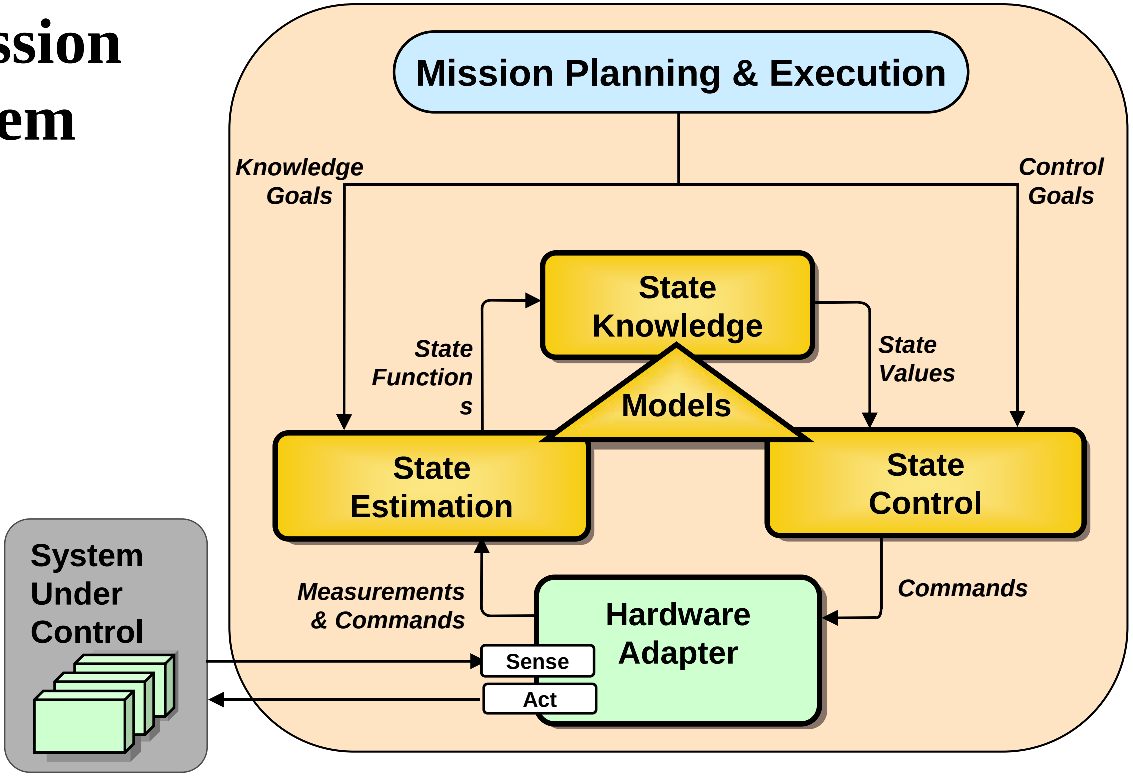
MER Ops personnel use MAPGEN to:

- Plan Activities (Goals)
- Analyze Resources
- Edit Plans

Benefits (1)

- Robustness:
  - Control layer has flexibility in achieving goal
  - Enables integration of tiered fault management capabilities

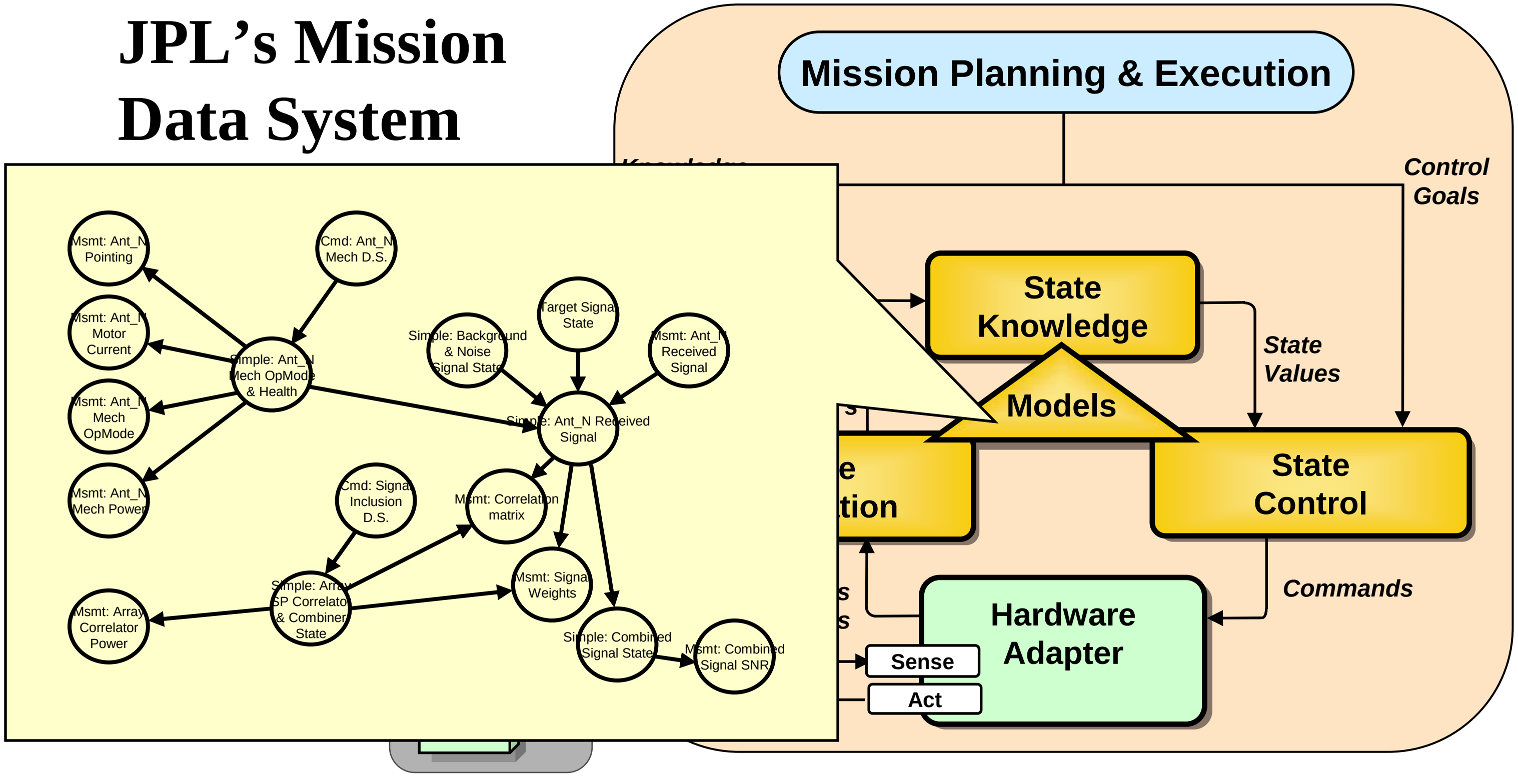
JPL's Mission Data System



Benefits (1)

- Robustness:
  - Control layer has flexibility in achieving goal
  - Enables integration of tiered fault management capabilities

JPL's Mission Data System



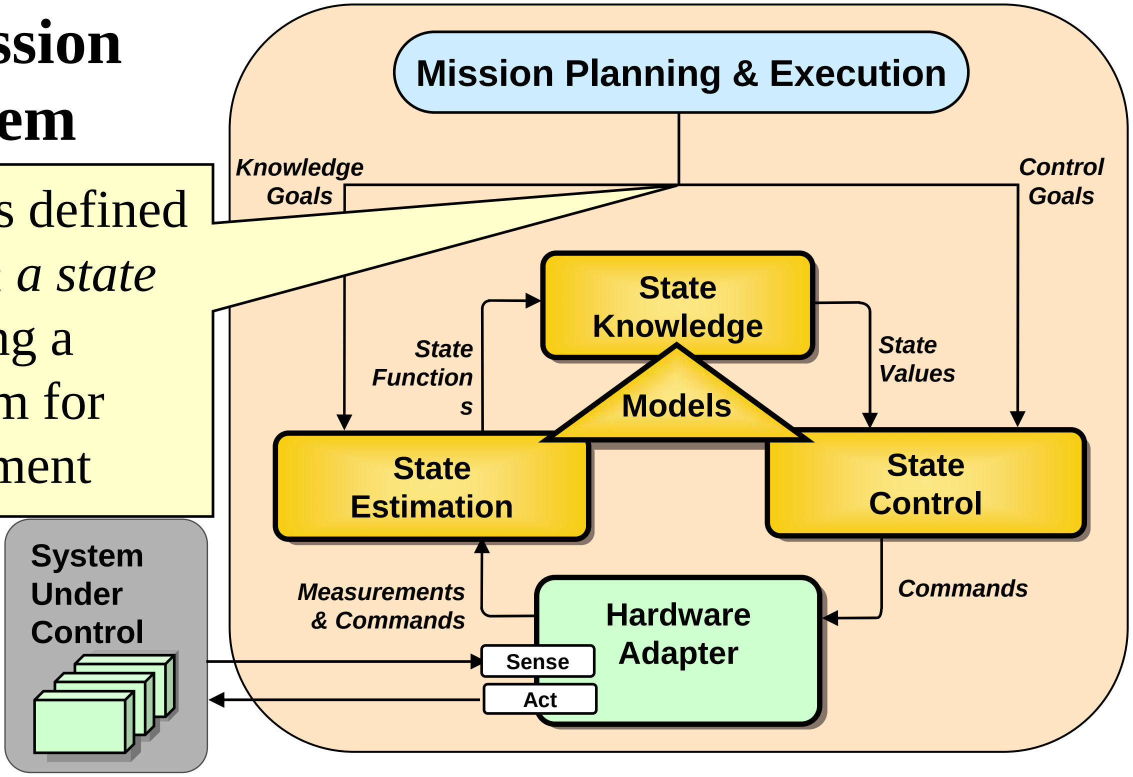
Mission Planning & Execution
Control Goals
State Knowledge
State Values
Models
State Control
Commands
Hardware Adapter
Sense
Act
Msmt: Ant_N Pointing
Cmd: Ant_N Mech D.S.
Msmt: Ant_N Motor Current
Simple: Ant_N Mech OpMode & Health
Msmt: Ant_N Mech OpMode
Msmt: Ant_N Mech Power
Target Signal State
Simple: Background & Noise Signal State
Msmt: Ant_N Received Signal
Simple: Ant_N Received Signal
Cmd: Signal Inclusion D.S.
Msmt: Correlation matrix
Simple: Array SP Correlator & Combiner State
Msmt: Array Correlator Power
Msmt: Signal Weights
Simple: Combined Signal State
Msmt: Combined Signal SNR

Benefits (1)

- Robustness:
  - Control layer has flexibility in achieving goal
  - Enables integration of tiered fault management capabilities

JPL’s Mission Data System

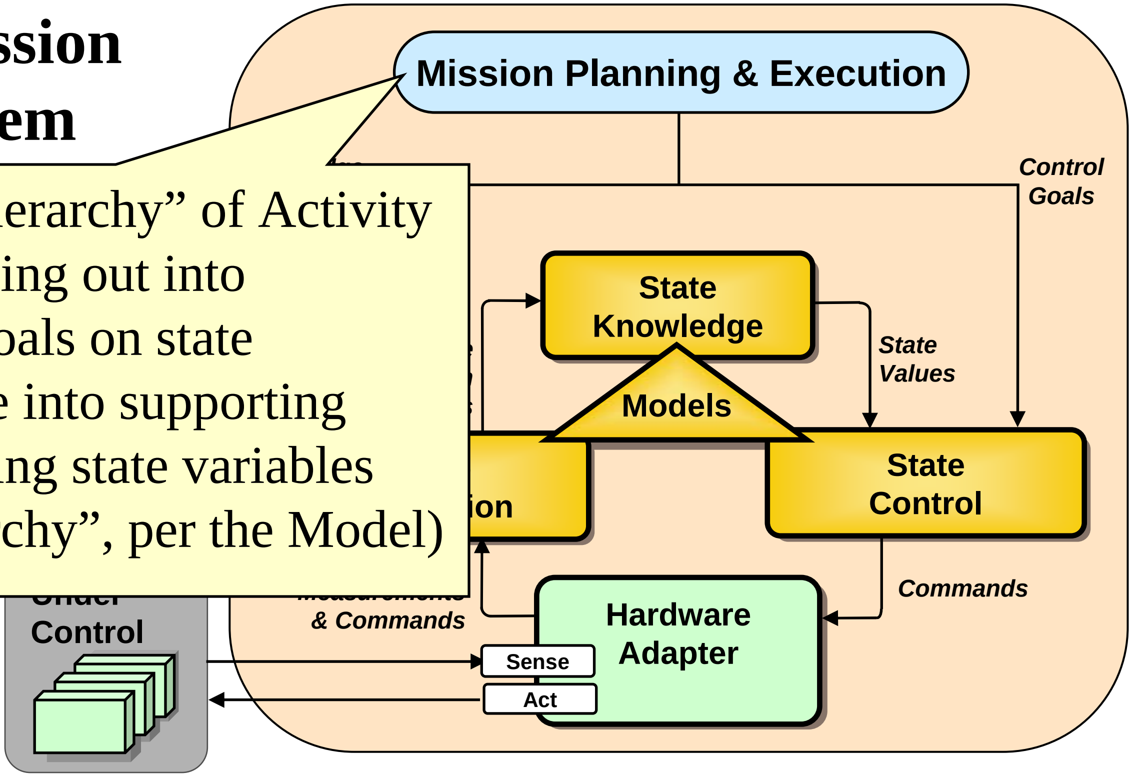
In MDS, a Goal is defined as a *constraint on a state variable*, providing a natural mechanism for resource management



Benefits (1)

- Robustness:
  - Control layer has flexibility in achieving goal
  - Enables integration of tiered fault management capabilities

JPL's Mission Data System



- "Abstraction hierarchy" of Activity Macros bottoming out into sequences of goals on state
- Goals elaborate into supporting goals on affecting state variables ("Causal hierarchy", per the Model)

Benefits (2)

- Lower Ops Costs
- Greater Science Return

Autonomous Sciencecraft Experiment on EO-1

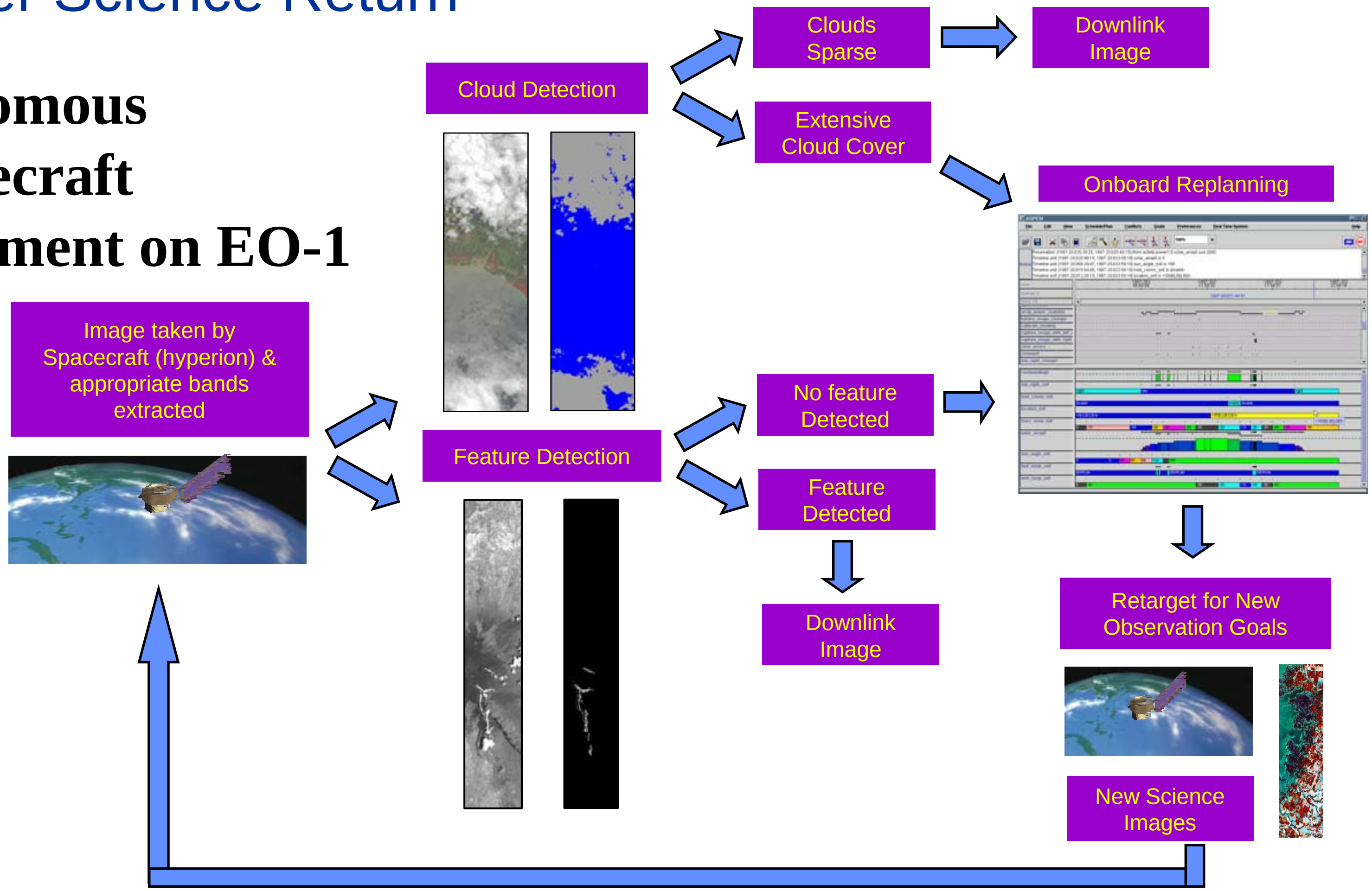


Image taken by Spacecraft (hyperion) & appropriate bands extracted

Cloud Detection

Clouds Sparse

Downlink Image

Extensive Cloud Cover

Onboard Replanning

Feature Detection

No feature Detected

Feature Detected

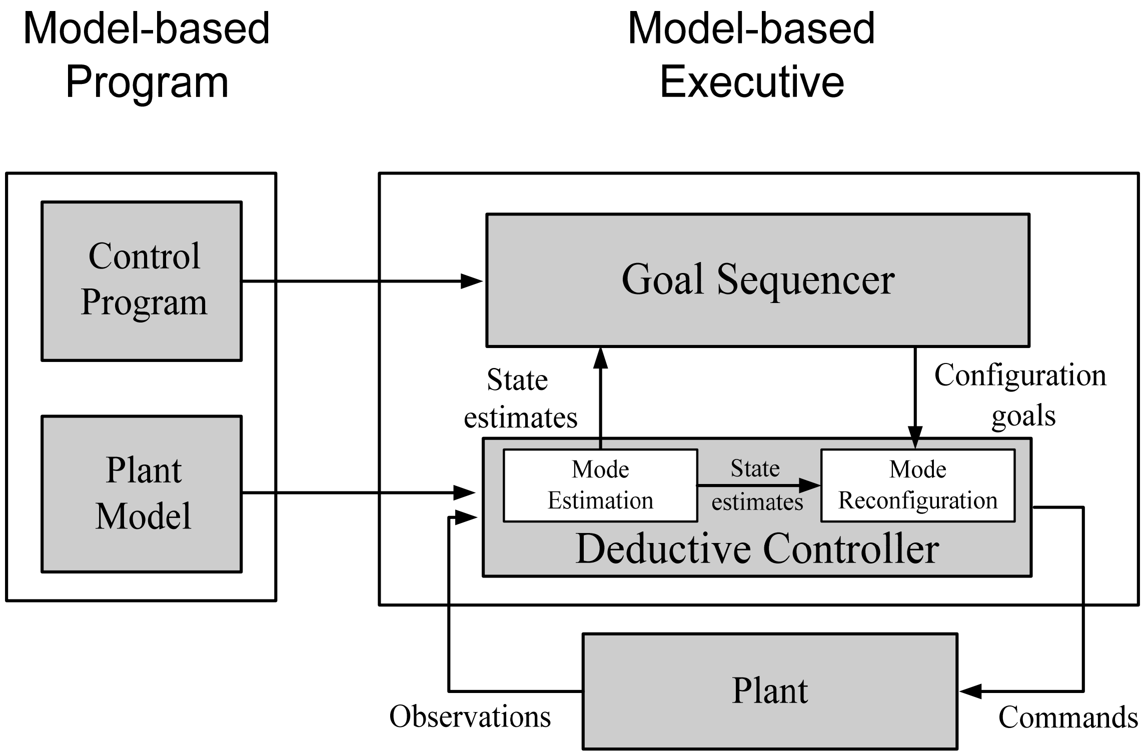
Downlink Image

Retarget for New Observation Goals

New Science Images

Benefits (3)

- Mission-enabling Autonomy:
  - via integration of state-of-the-art model-based software technologies

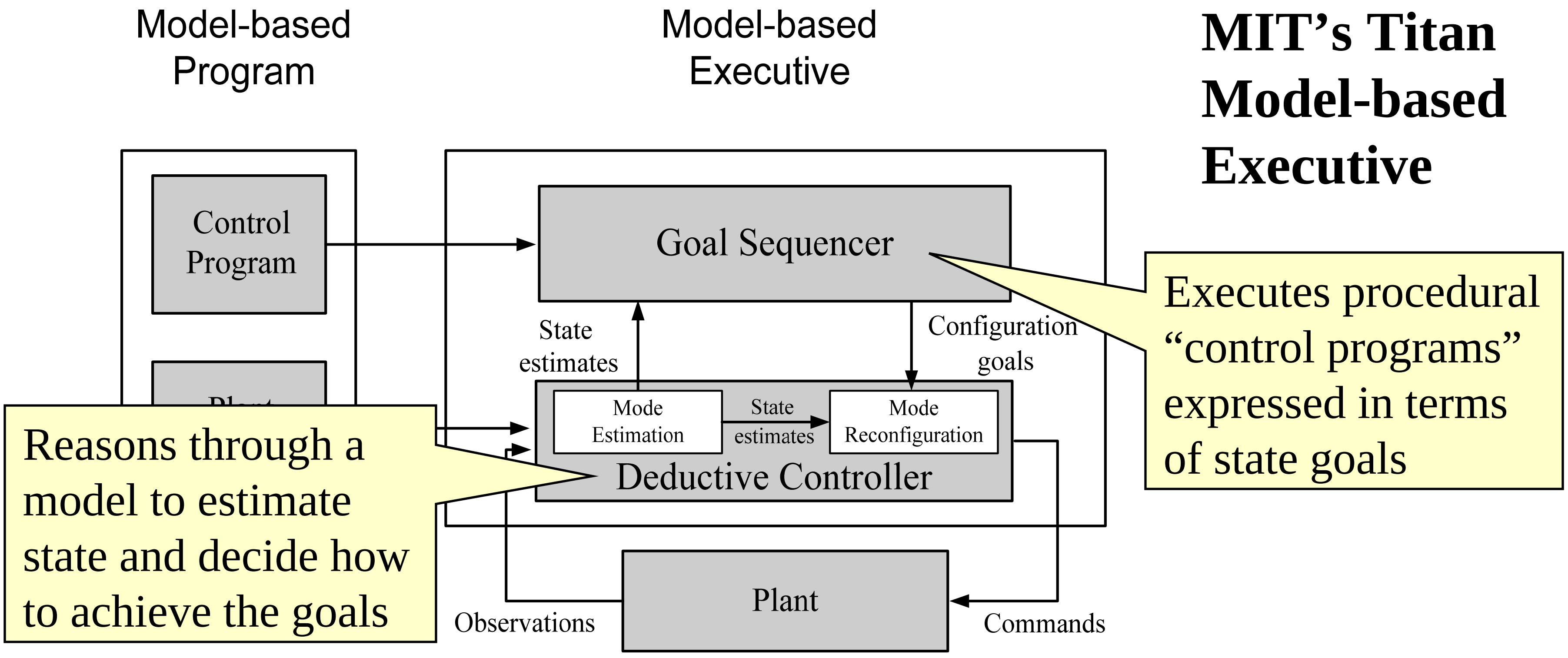


MIT's Titan Model-based Executive

Benefits (3)

- Mission-enabling Autonomy:
  - via integration of state-of-the-art model-based software technologies

MIT's Titan Model-based Executive



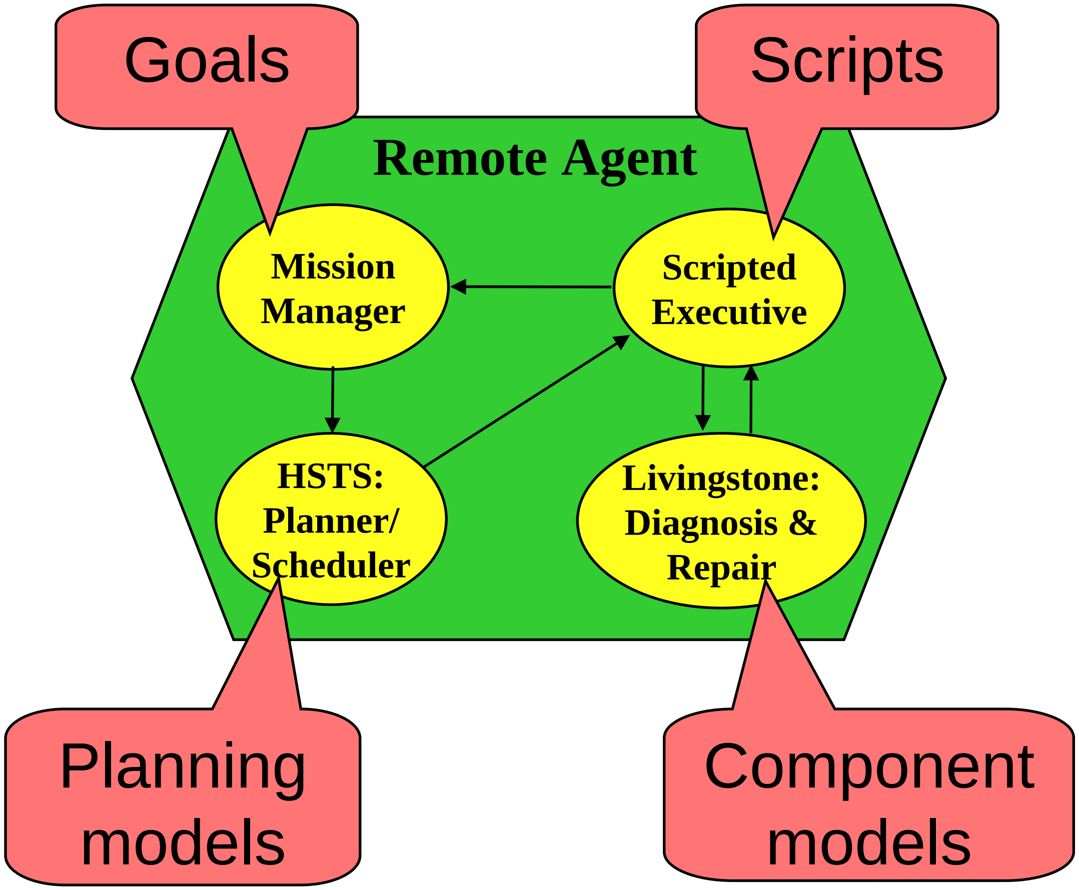
Executes procedural "control programs" expressed in terms of state goals

Reasons through a model to estimate state and decide how to achieve the goals

Challenge Questions

- How do we avoid the potential for divergence and knowledge duplication due to use of multiple knowledge representations?
- How can we facilitate transitioning the operational paradigm from “product flow” to “work flow”?
- How do we design for operability (i.e., integrate goal-based operations into the end-to-end mission lifecycle)?
- Can we adapt legacy tools to this new operations paradigm?
- How can we assure the reliability of goal-based ops (V&V of goal-based ops tools)?
- How do we overcome the “cultural” hurdles to acceptance of these new methods and tools?

Multiplicity of knowledge representations



- Different modules require distinct knowledge representation
  - benefit: ability to reason at different levels of abstraction
  - drawbacks: potential divergent models, knowledge duplication

Multiplicity of knowledge representations

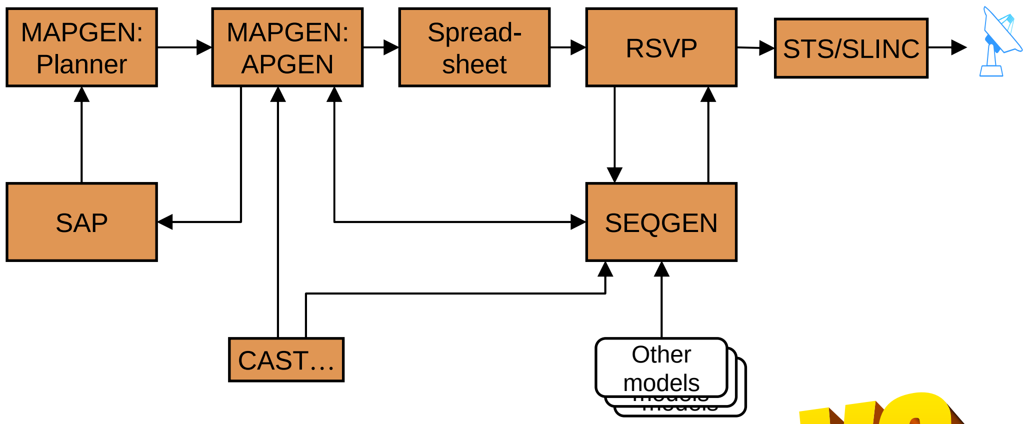
Barrier to wide deployment of autonomy s/w:

numerous tasks use variety of
modeling & programming languages

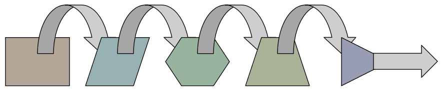
The Challenge:

- ✓ head toward unified representation of spacecraft
- ✓ accommodate complexities of spacecraft domain
- ✓ maintain capacity for knowledge abstraction

Transitioning from "product flow" to "work flow"

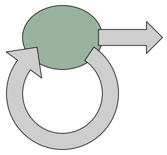


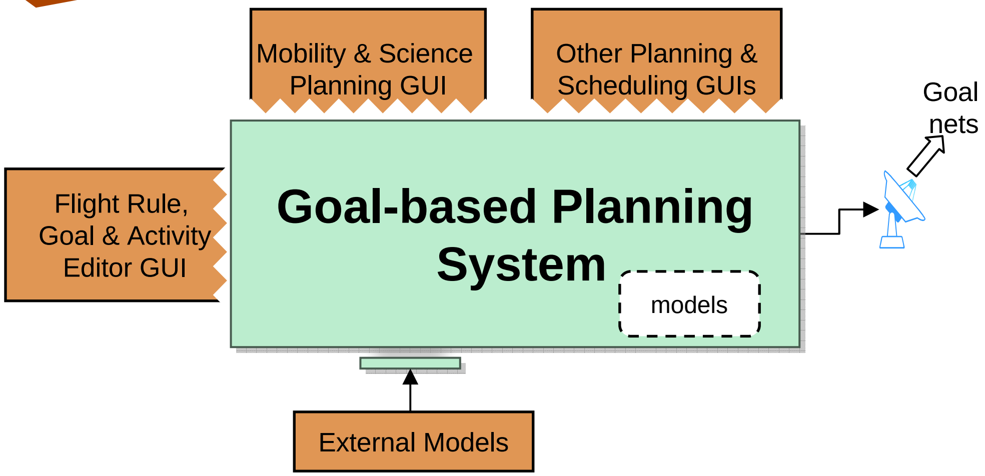
Product Flow



VS.

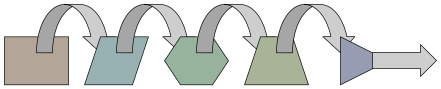
Work Flow



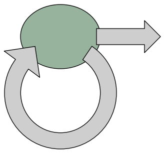

Mobility & Science Planning GUI
Other Planning & Scheduling GUIs
Goal nets
Flight Rule, Goal & Activity Editor GUI
Goal-based Planning System
models
External Models

Transitioning from "product flow" to "work flow"

- Goal-based operations facilitates a shift in our approach:
  - From product flow

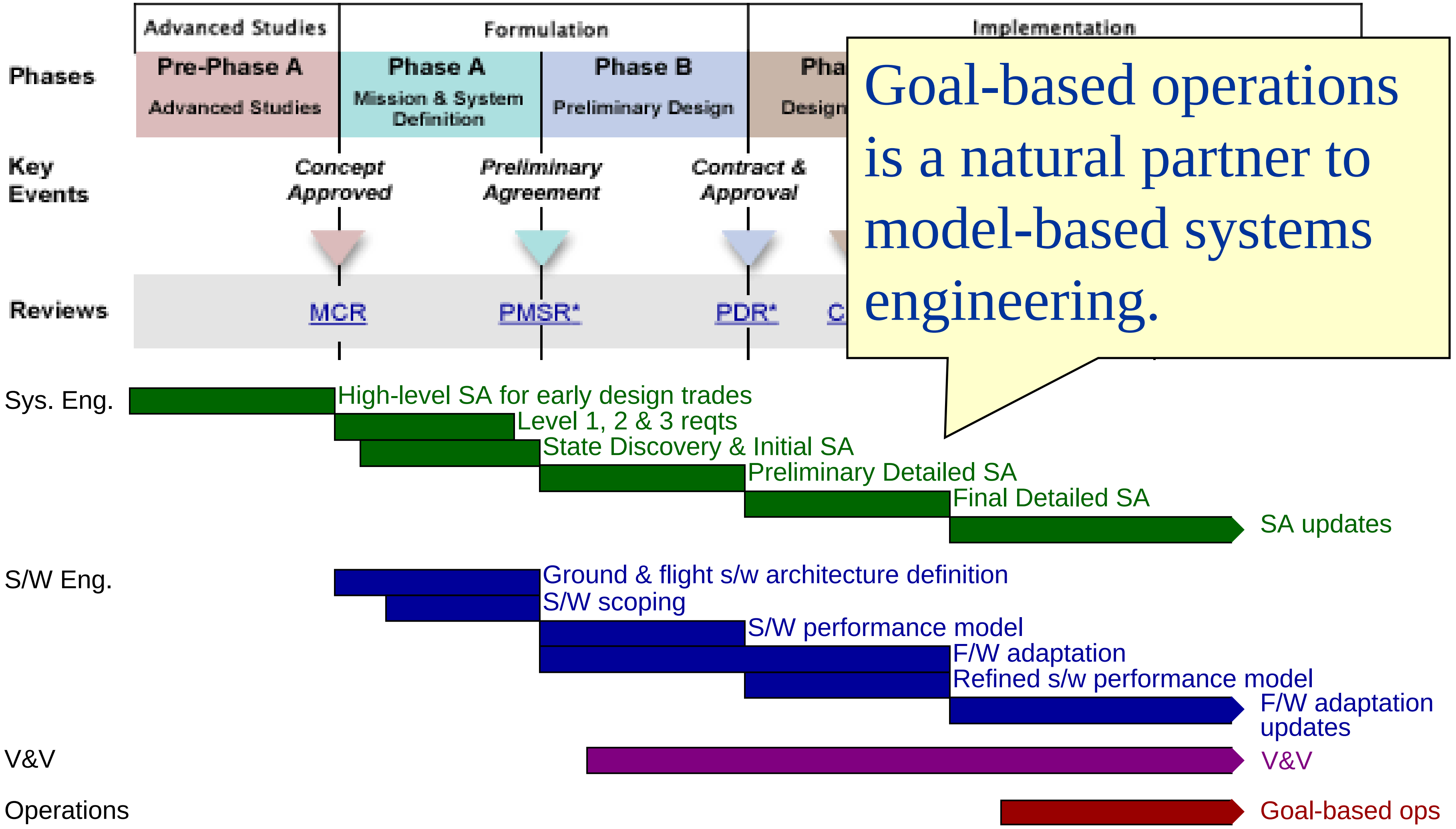


    - Development progressing from one tool to another through exchange of data files along a development path
    - Progress is measured by where activity is in the tool chain
    - Reverse flow to address problems is awkward, at best, and usually avoided
      - Fixes often made in place without benefit of earlier steps
  - To work flow



    - One uniform product set managed by a common tool going through successive stages of refinement
    - Progress is measured by level of completeness, validation, and approval
      - Manageable through a parallel workflow process
    - Reversing to address problems is straightforward

Integration of goal-based ops into the mission lifecycle



Sys. Eng.

High-level SA for early design trades

Level 1, 2 & 3 reqts

State Discovery & Initial SA

Preliminary Detailed SA

Final Detailed SA

SA updates

S/W Eng.

Ground & flight s/w architecture definition

S/W scoping

S/W performance model

F/W adaptation

Refined s/w performance model

F/W adaptation updates

V&V

V&V

Operations

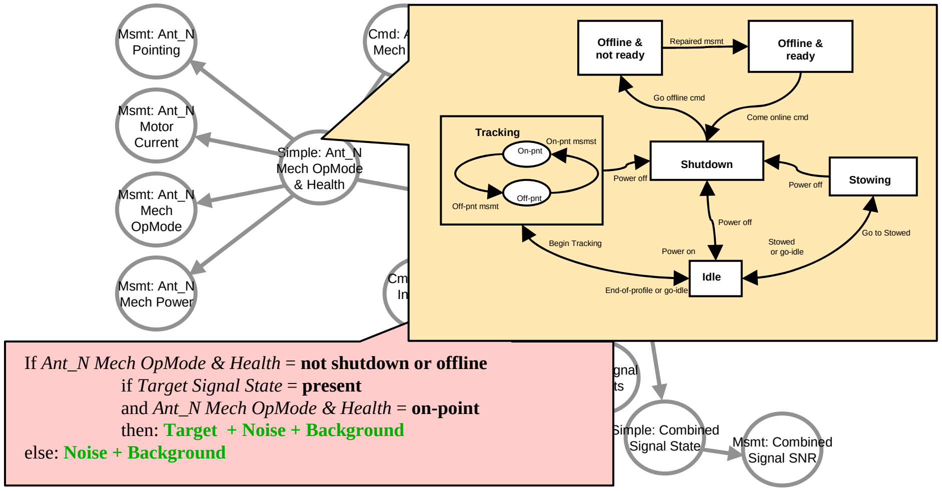
Goal-based ops

Goal-based operations is a natural partner to model-based systems engineering.

Integration of goal-based ops into the mission lifecycle

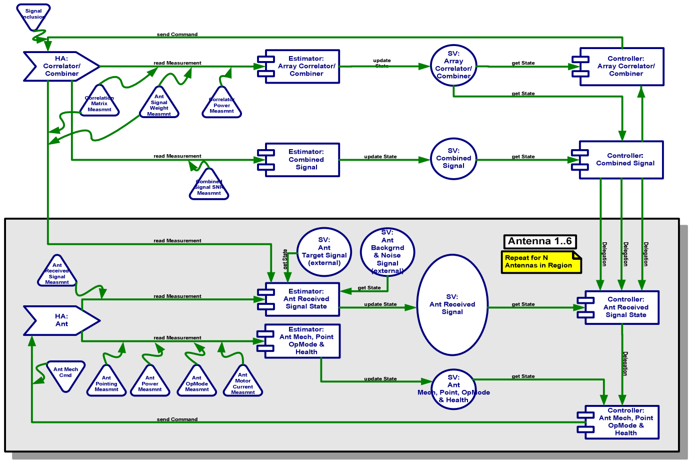
1. System to be controlled

2. State Analysis produces model



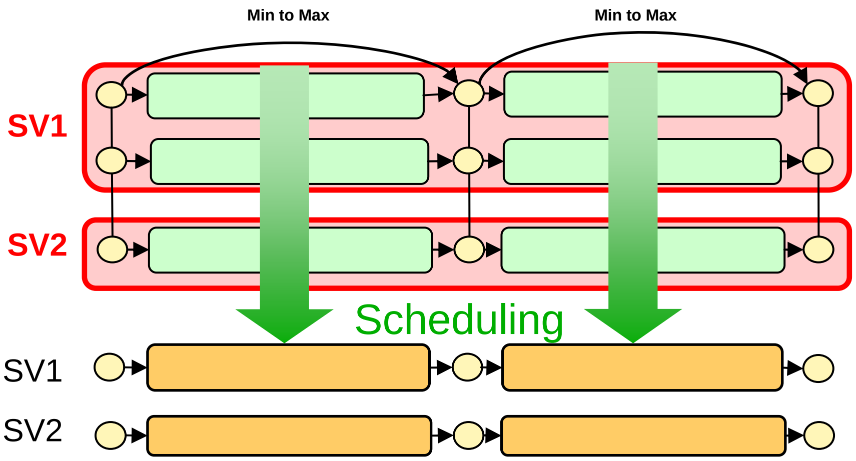
If *Ant_N Mech OpMode & Health* = **not shutdown or offline**
if *Target Signal State* = **present**
and *Ant_N Mech OpMode & Health* = **on-point**
then: **Target + Noise + Background**
else: **Noise + Background**

3. Model informs software design

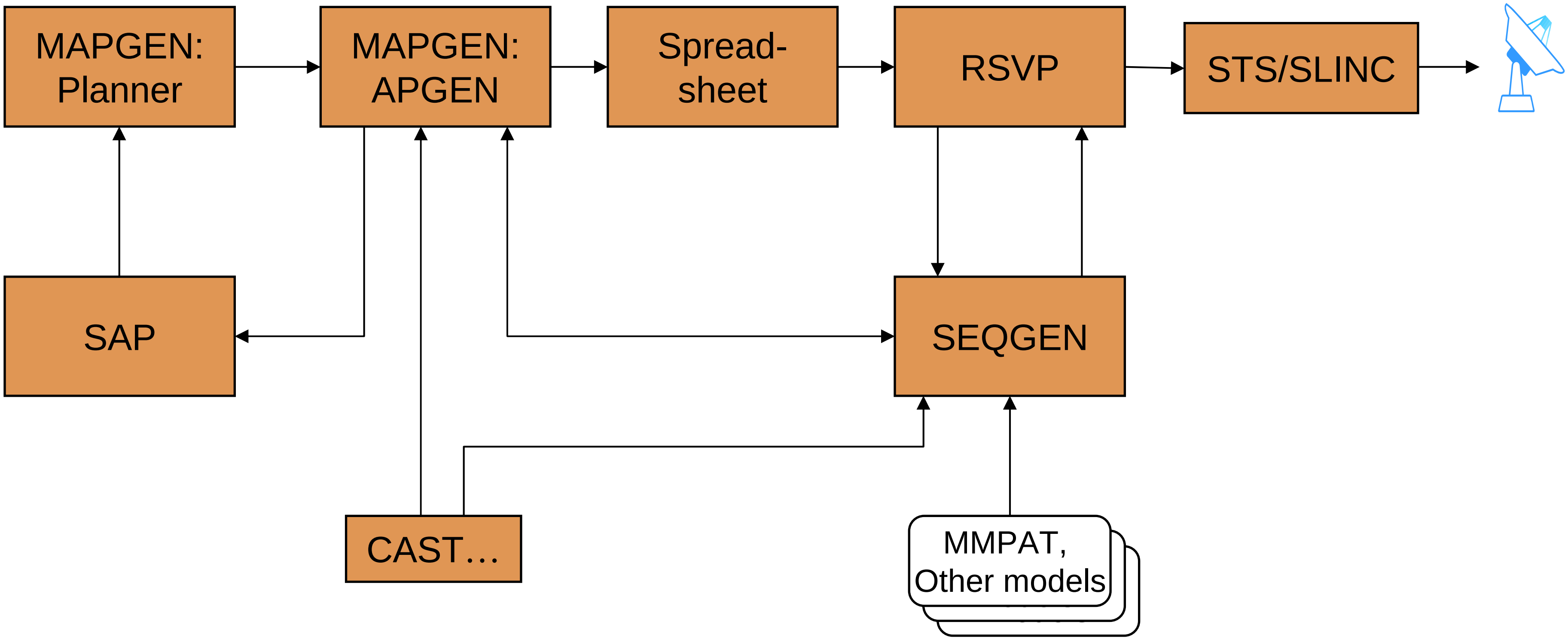

Signal Inclusion
send Command
HA: Correlator/ Combiner
read Measurement
Estimator: Array Correlator/ Combiner
update State
SV: Array Correlator/ Combiner
get State
Controller: Array Correlator/ Combiner
Correlation Matrix Measmnt
Ant Signal Weight Measmnt
Correlator Power Measmnt
get State
read Measurement
Estimator: Combined Signal
update State
SV: Combined Signal
get State
Controller: Combined Signal
Combined Signal SNR Measmnt
Delegation
Delegation
Delegation
read Measurement
get State
SV: Ant Target Signal (external)
SV: Ant Backgrnd & Noise Signal (external)
Antenna 1..6
Repeat for N Antennas in Region
Ant Received Signal Measmnt
read Measurement
Estimator: Ant Received Signal State
get State
update State
SV: Ant Received Signal
get State
Controller: Ant Received Signal State
HA: Ant
read Measurement
Estimator: Ant Mech, Point OpMode & Health
Delegation
Ant Mech Cmd
Ant Pointing Measmnt
Ant Power Measmnt
Ant OpMode Measmnt
Ant Motor Current Measmnt
update State
SV: Ant Mech, Point, OpMode & Health
get State
send Command
Controller: Ant Mech, Point OpMode & Health

DSAN Collaboration Diagram

4. Model informs operations



Adapting legacy tools



Can we re-architect the software, but leverage the existing tools' functionality, while providing familiar/comfortable user interfaces?

V&V of goal-based ops tools

- Comprehensive V&V plan:
  - Engine & Model validation
  - High-fidelity mission testbeds
  - Auto-code generation where practical
  - Formal V&V methods where appropriate
- Where possible, initial flight validation on spacecraft with more aggressive risk posture
  - Technology validation missions (e.g., New Millennium Program)
  - Post-primary mission spacecraft assets
- Progressive capability phasing
- Ground-to-flight migration of capabilities
- Design for variable autonomy
- Extended deployments and in-situ stress testing

Cultural hurdles to acceptance

- Part of this is a “trust” issue, somewhat related to the previous challenge question
- This issue applies more broadly to any new technology, *especially* software technology
- “If it hasn’t flown before, I don’t want to fly it” - what incentives are there for Project Managers to embrace (or at least accept) new technology? This is an organizational challenge…

Motivating a Standard for Goal-based Operations

Michel D. Ingham, Sc.D.

Jet Propulsion Laboratory
California Institute of Technology

Ground System Architecture Workshop 2006

"Toward a Standard for Goal-Based Operations" Working Group

Manhattan Beach, CA

March 29, 2006

Why bother with a Standard?

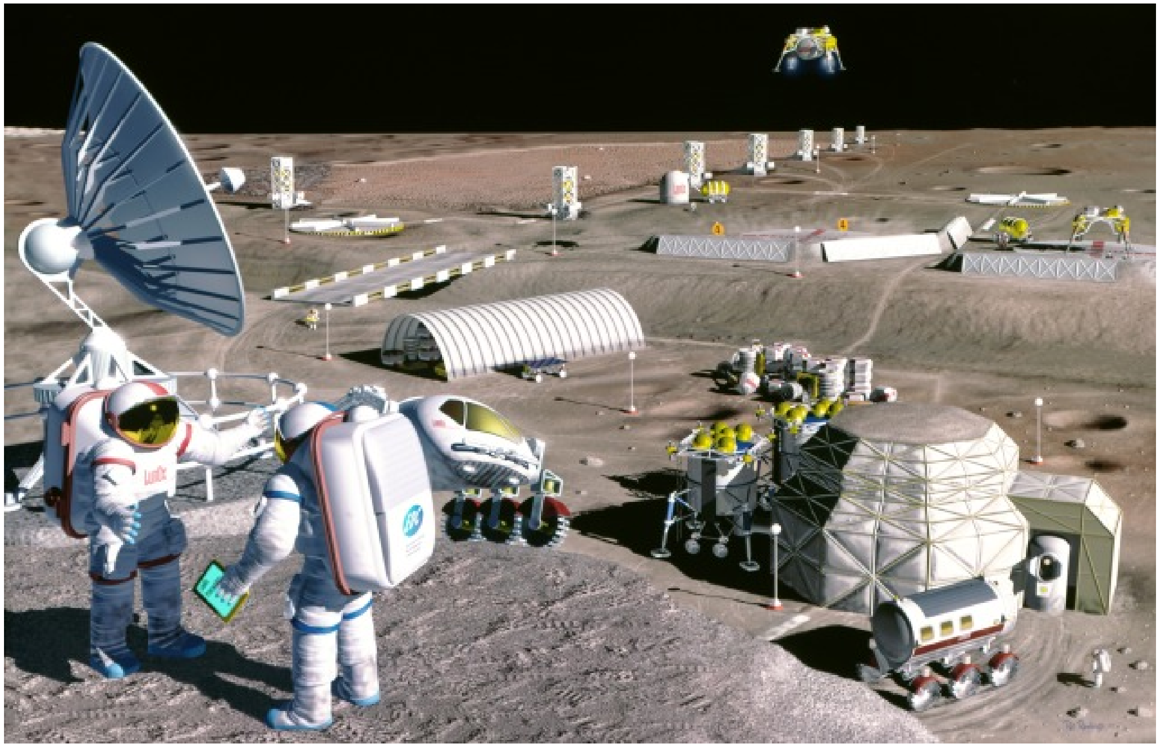
- Current ops approach doesn't really have a "standard", does it?
  - Each space-faring organization has its own accepted command language and set of ops processes

Because…

- New classes of mission, requiring significantly greater reuse and interoperability, are pushing towards an ops standard (whether goal-based or command-based)

Because…

- Huge endeavors like Project Constellation will be accomplished by many different organizations – can we safely assume that the disparate elements will be fully interoperable without enforcing a Standard?



But still…

- Developing a good Standard takes time
  - Probably shouldn’t rush it, and risk missing the mark

- In the near term, can probably make significant strides in promoting wider acceptance of the Goal-based Ops approach, even in the absence of a Standard
  - Will require ***greater discipline*** than we’ve shown in the past to really ARCHITECT the system
  - I’m talking about ***integrated architecture***: of the spacecraft, of the ground system, of the operations approach…

- The trick will be to bring the Standard online before too many “bad habits” have been formed!

What belongs in the Standard?

- Acceptable representation(s) for intent?
- General form(s) for event-driven sequences (i.e., flexible time representation)?
- Ops Process?
- V&V Process?
- Human interface requirements?
- “Adjustable Autonomy” guidelines?
- Planning, scheduling and/or execution semantics? (probably not)

What type of Standard?

- Formal Standard, like Mil Specs?
  - Will require time and money. Who would foot the bill?
- Defacto Standard, like Linux?
  - Can we count on natural evolution to result in convergence?