



**JPL**

# 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



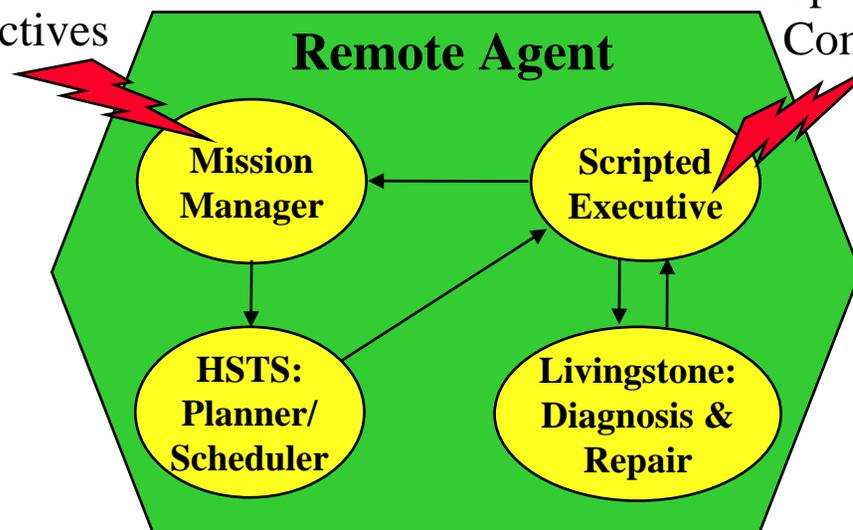
# Steps in the Right Direction (1)

JPL



Mission Objectives

Spacecraft Commands





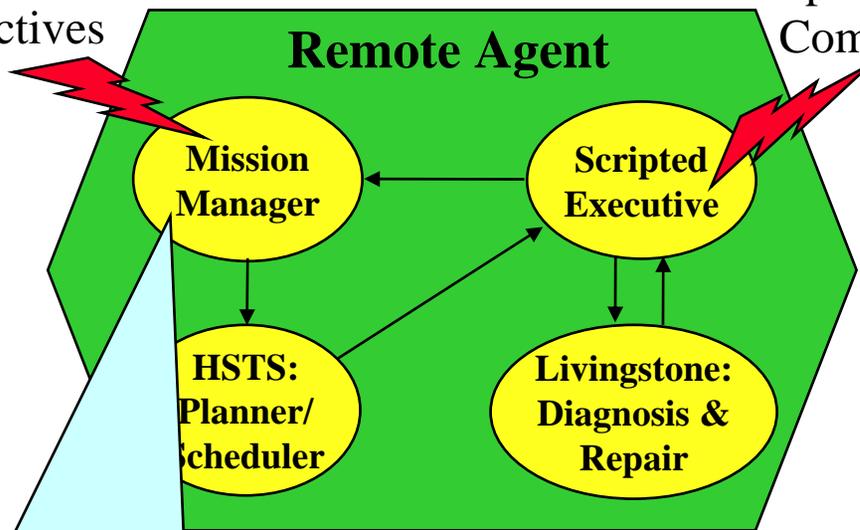
# Steps in the Right Direction (1)

JPL



Mission Objectives

Spacecraft Commands



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



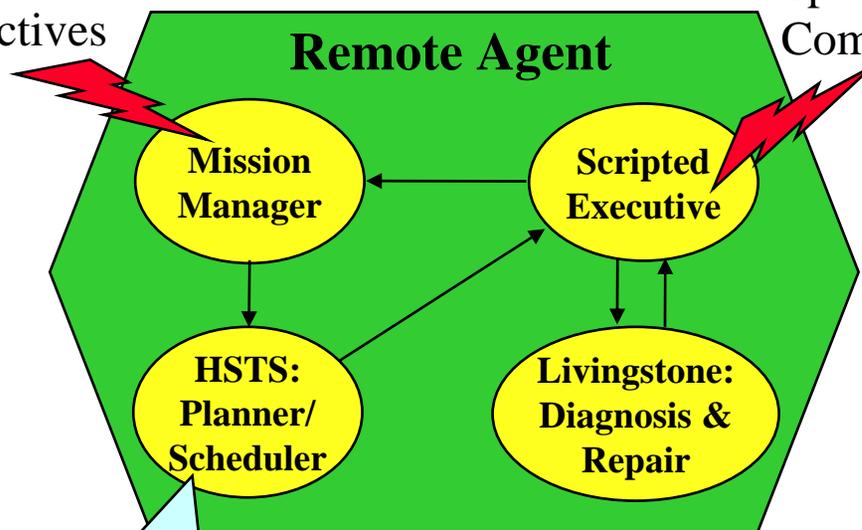
# Steps in the Right Direction (1)

JPL



Mission Objectives

Spacecraft Commands



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



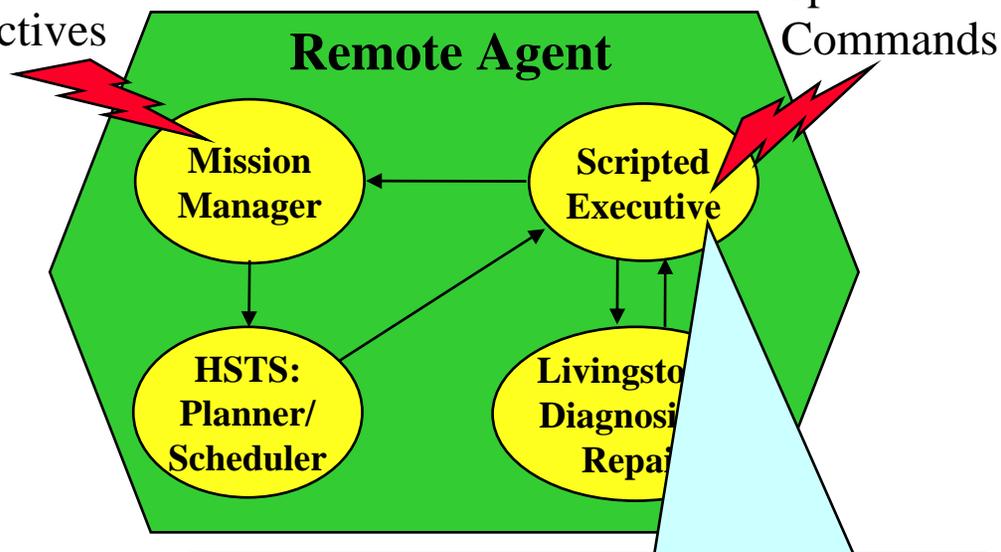
# Steps in the Right Direction (1)

JPL



Mission Objectives

Spacecraft Commands



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



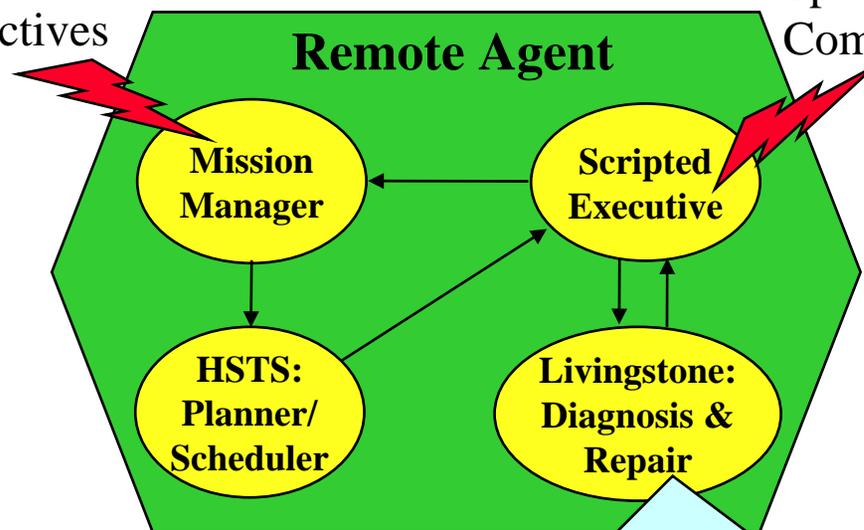
# Steps in the Right Direction (1)

JPL



Mission Objectives

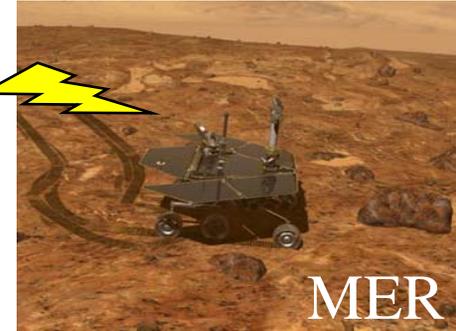
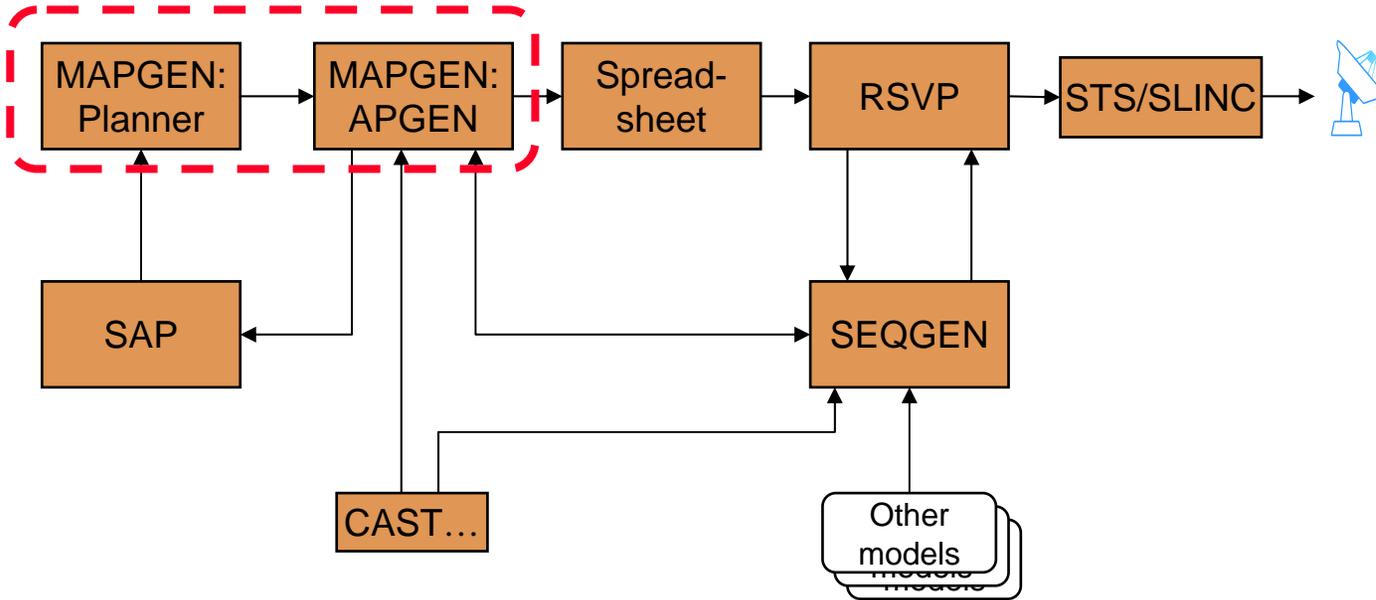
Spacecraft Commands



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

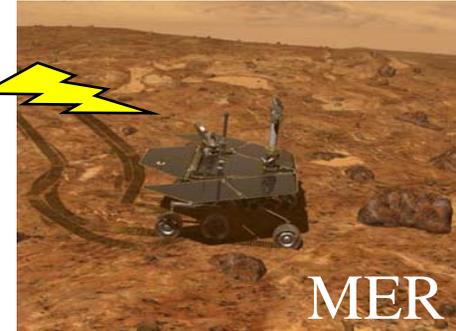
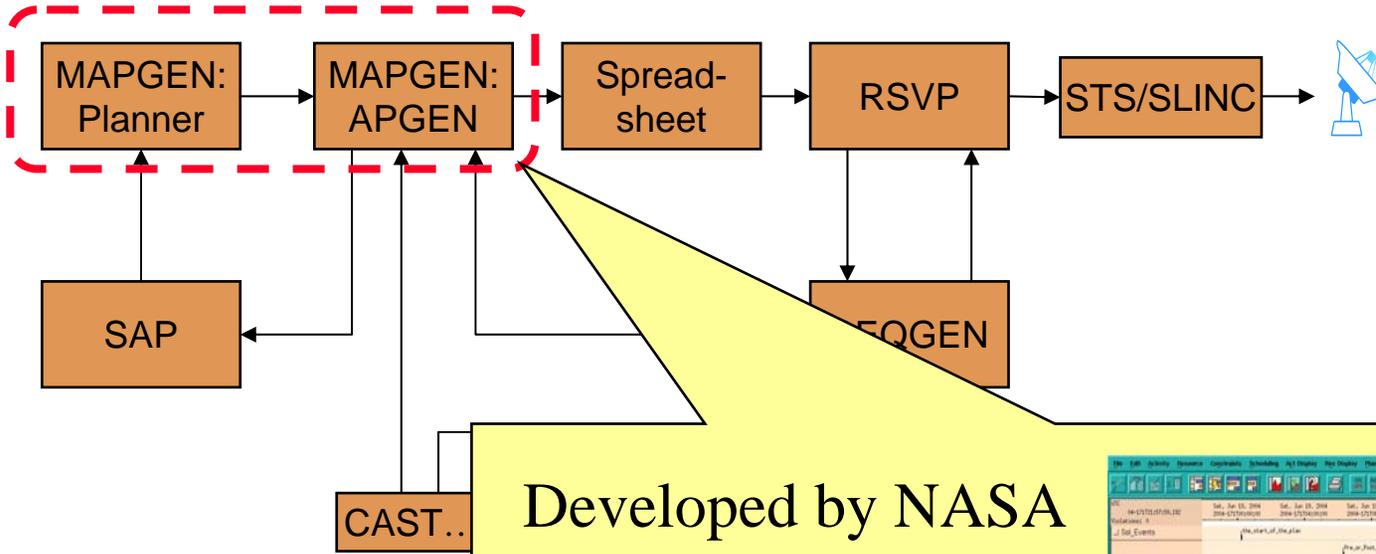


# Steps in the Right Direction (2)





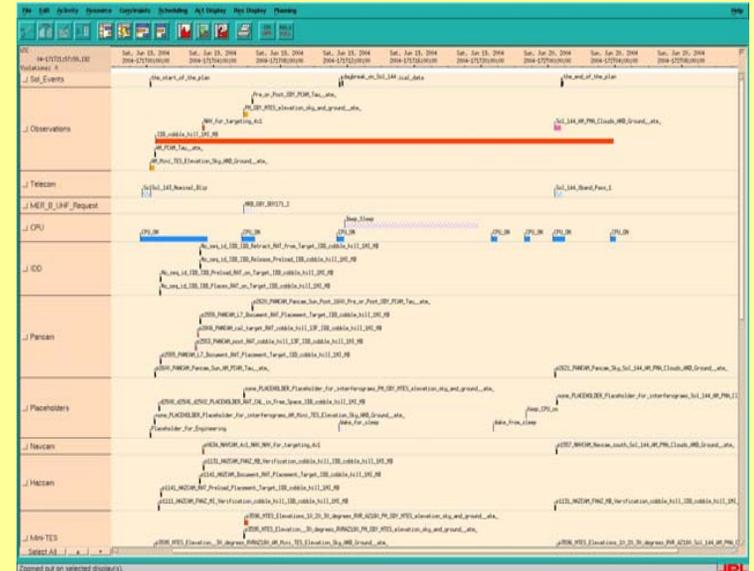
# Steps in the Right Direction (2)



Developed by NASA  
ARC & JPL;

MER Ops personnel  
use MAPGEN to:

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



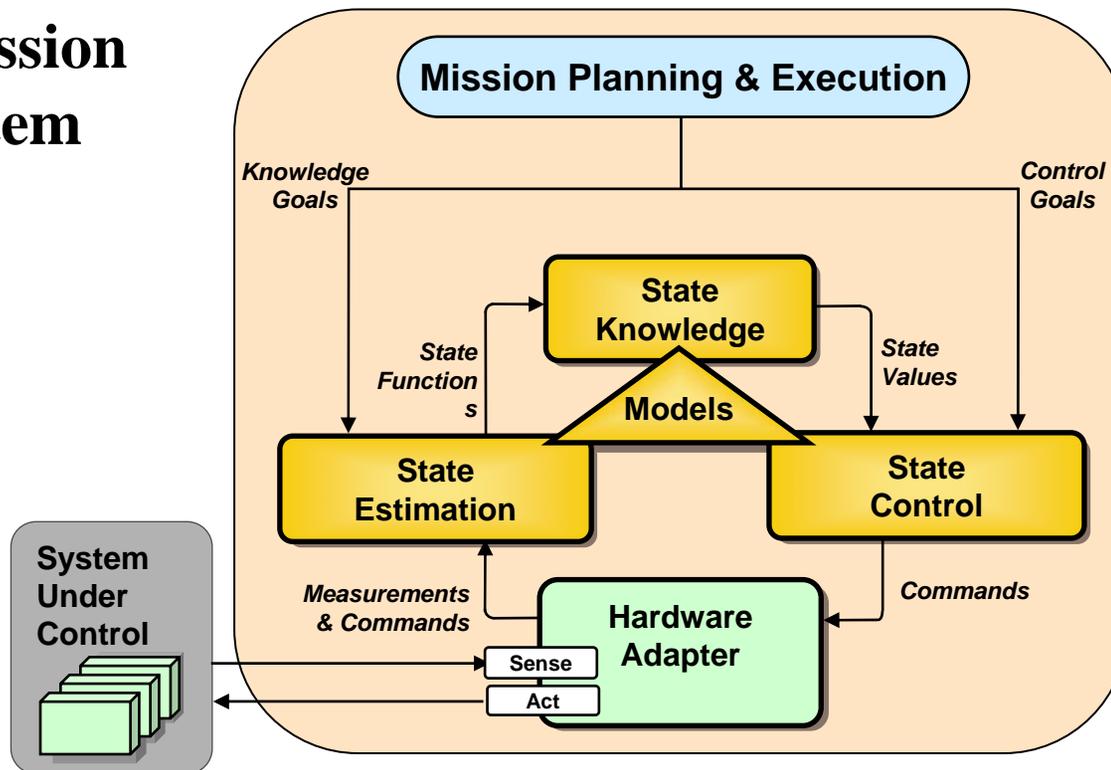


# Benefits (1)



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

## JPL's Mission Data System



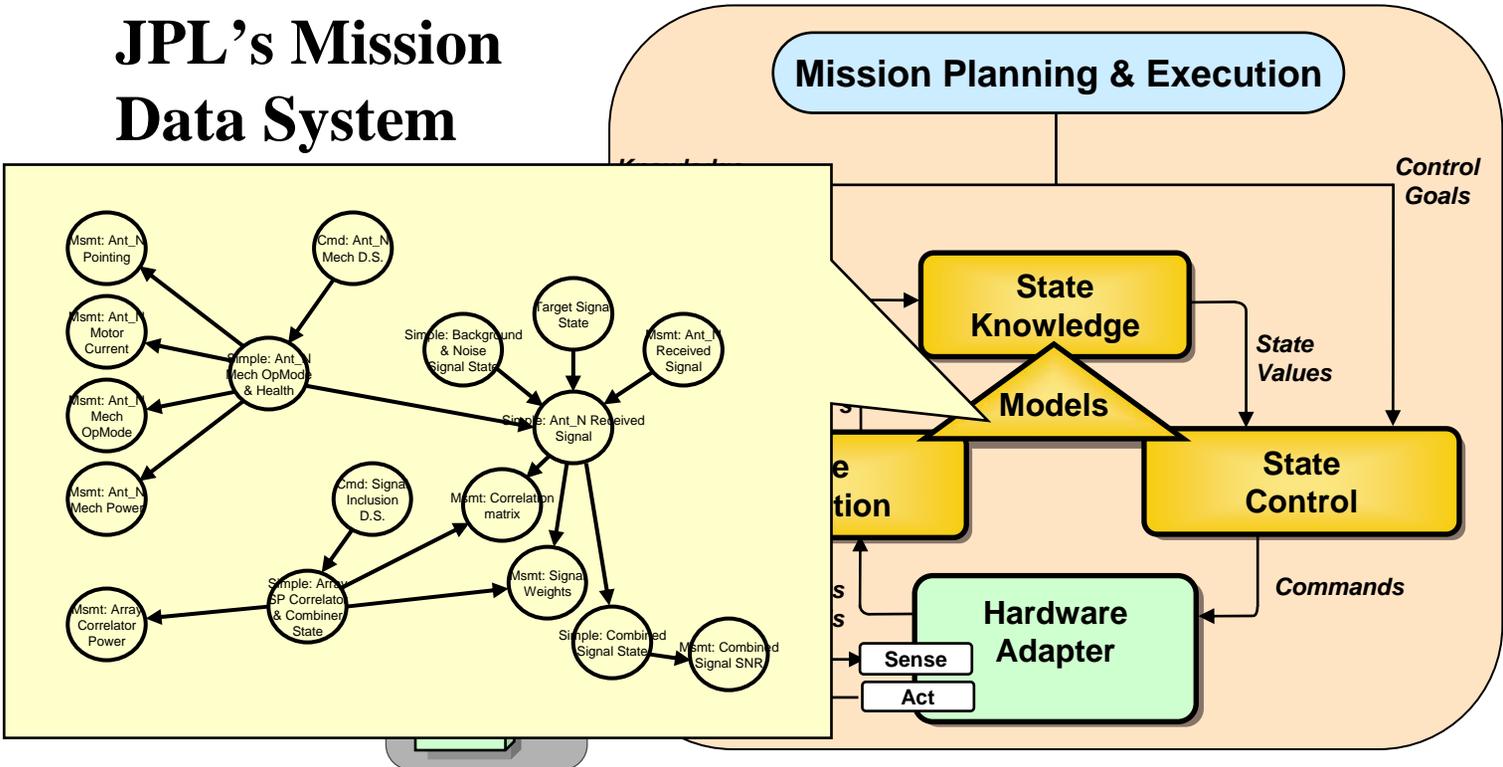


# Benefits (1)



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

## JPL's Mission Data System





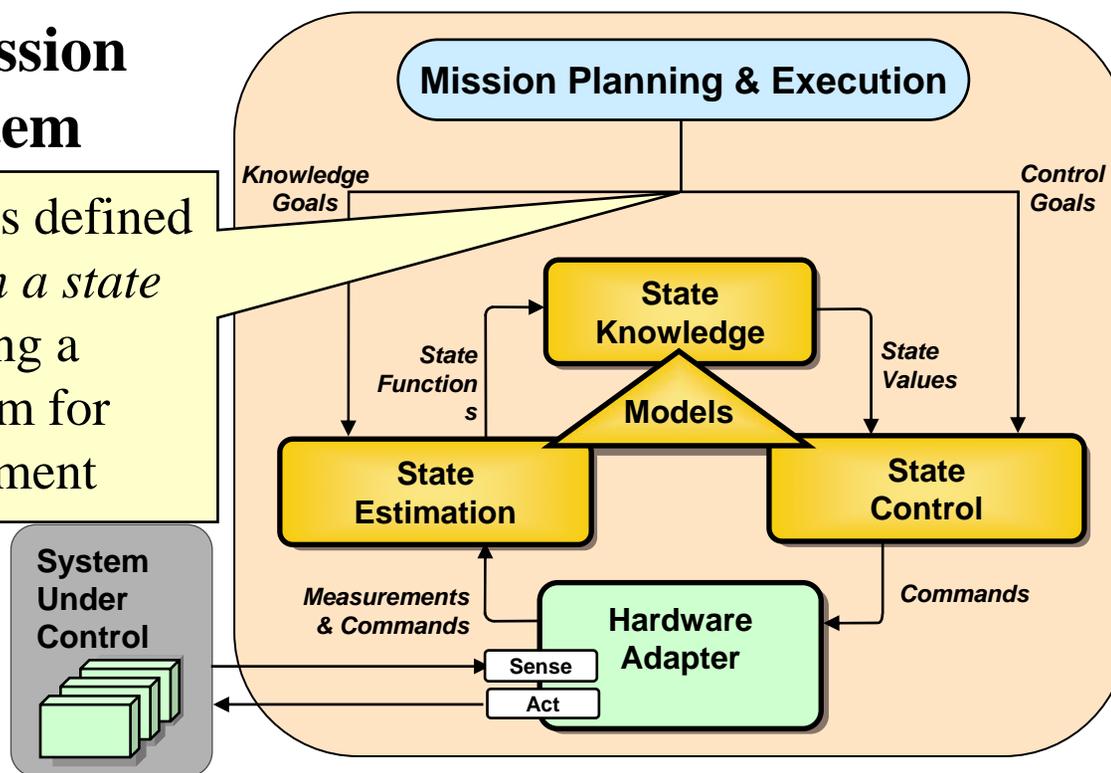
# Benefits (1)



- 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





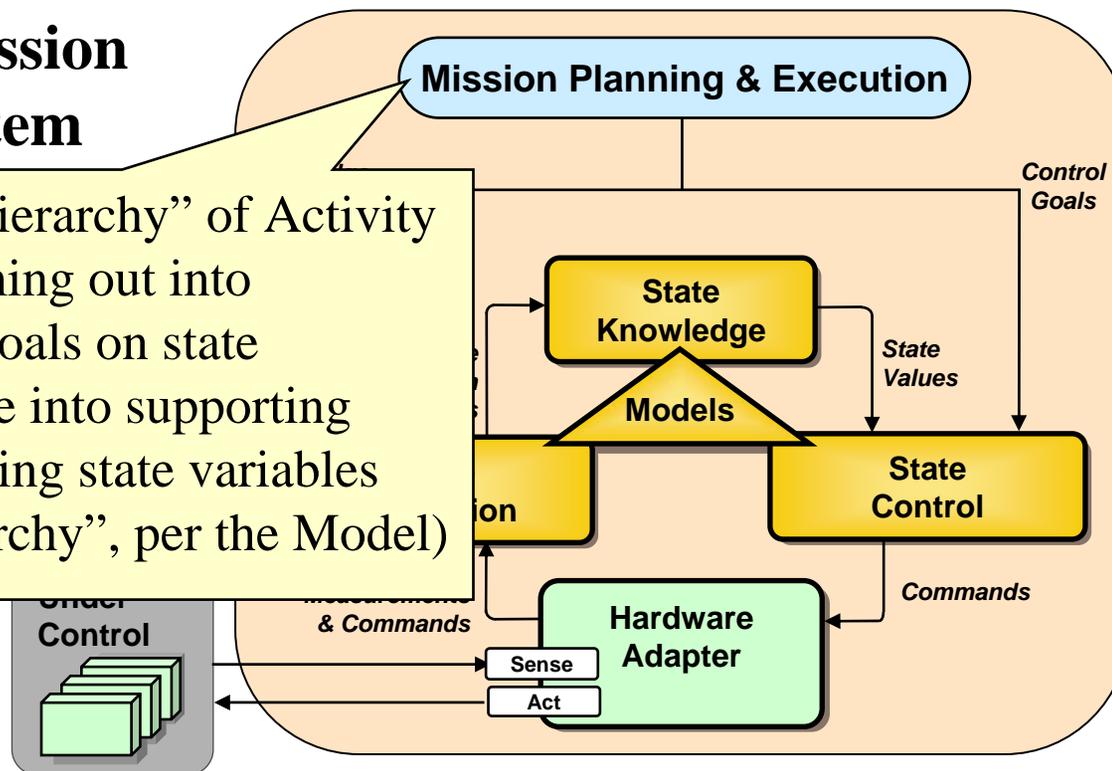
# Benefits (1)



- 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)



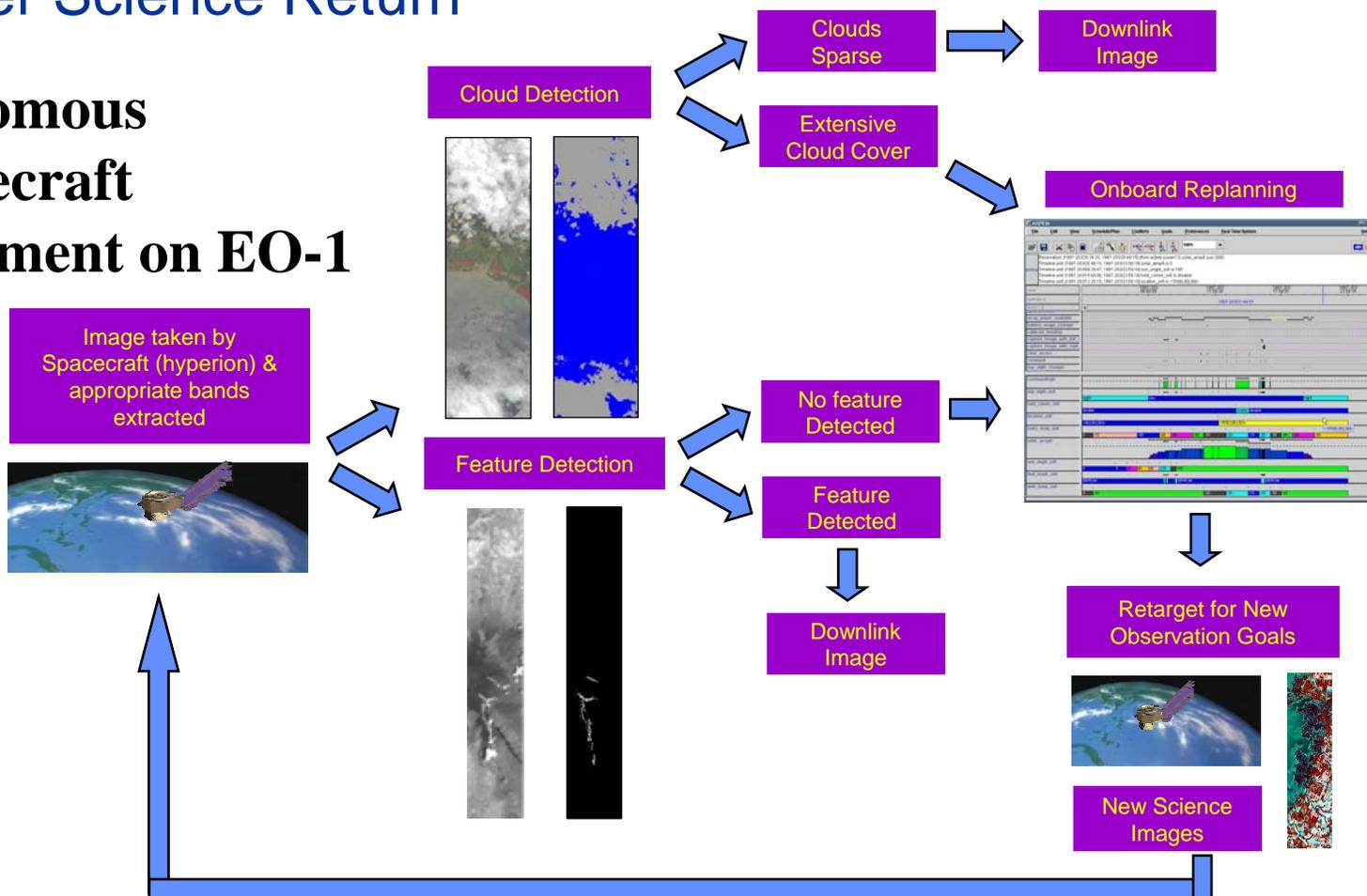


# Benefits (2)

**JPL**

- Lower Ops Costs
- Greater Science Return

## Autonomous Sciencecraft Experiment on EO-1



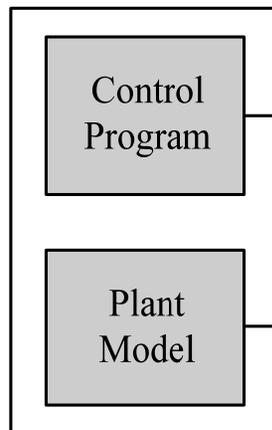


# Benefits (3)

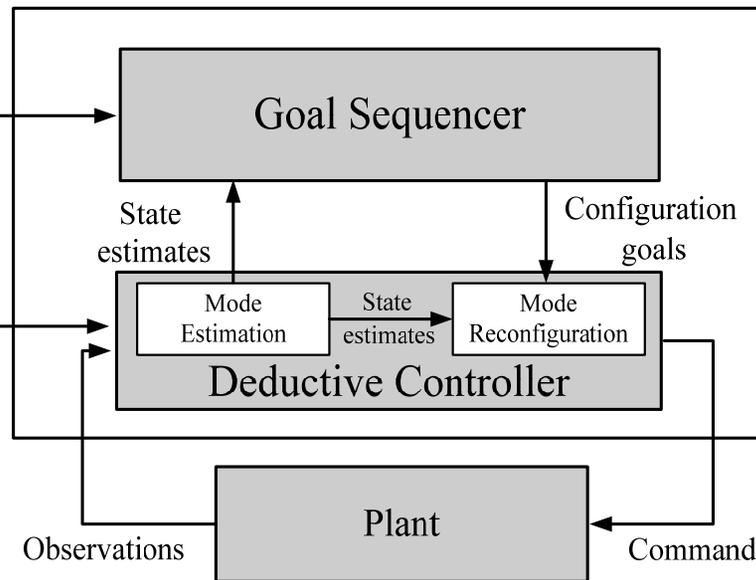


- Mission-enabling Autonomy:
  - via integration of state-of-the-art model-based software technologies

Model-based Program



Model-based Executive



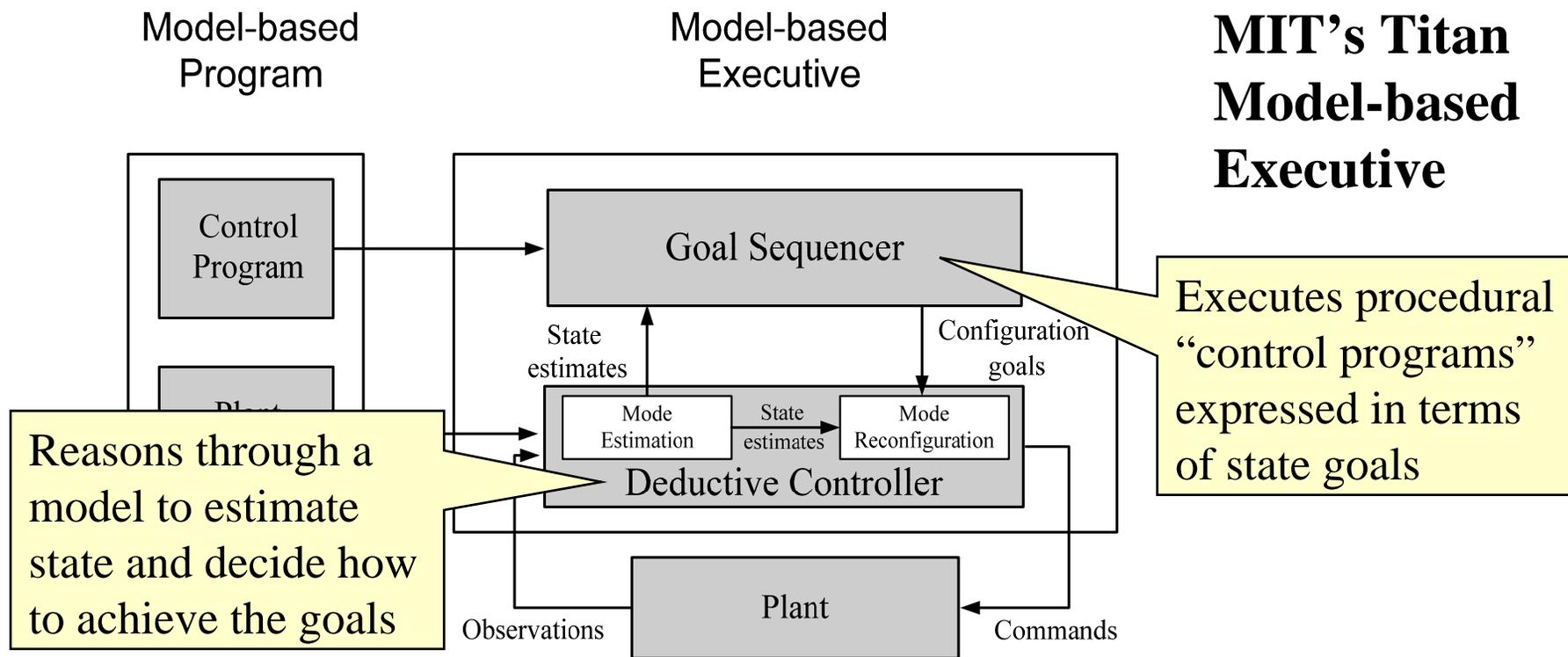
**MIT's Titan Model-based Executive**



# Benefits (3)

JPL

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





# Challenge Questions

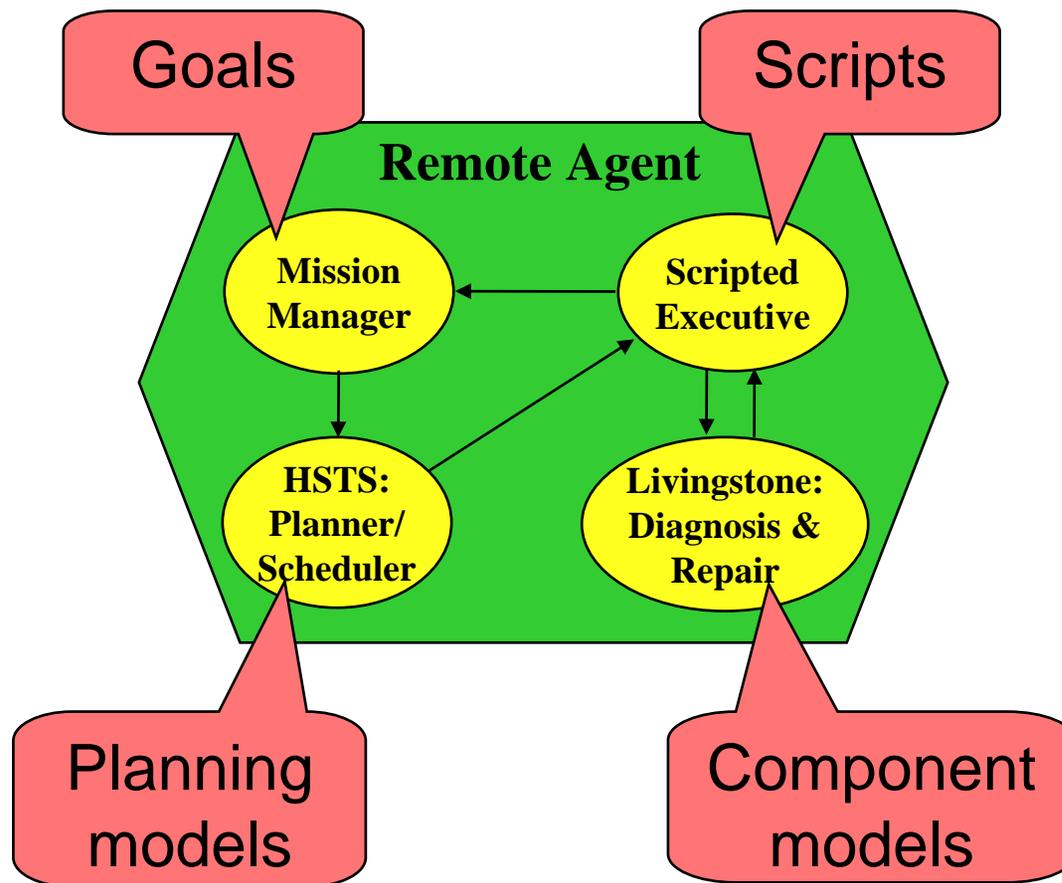
**JPL**

- 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?



**JPL**

# 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

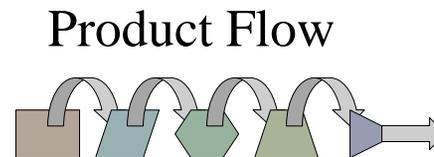
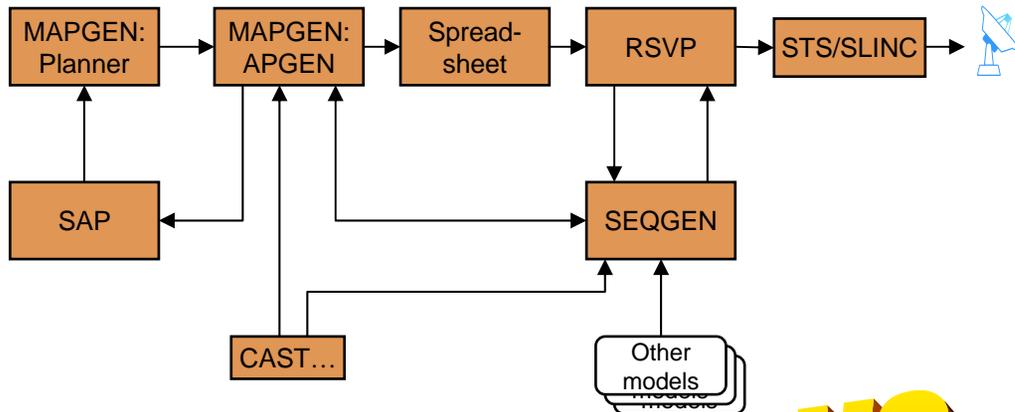
The Challenge:

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



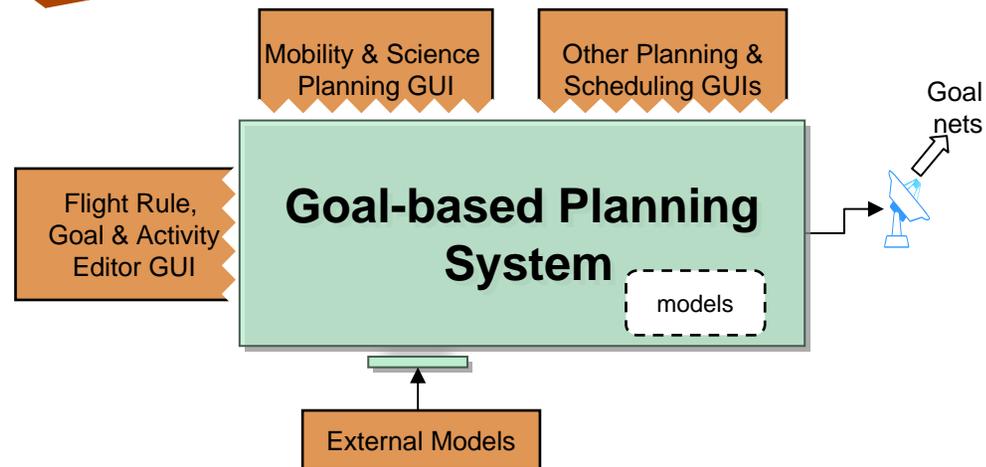
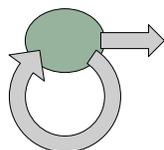
JPL

# Transitioning from “product flow” to “work flow”



# VS.

## Work Flow



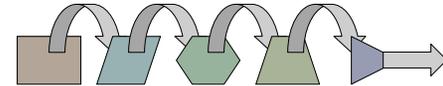


# Transitioning from “product flow” to “work flow”

- Goal-based operations facilitates a shift in our approach:

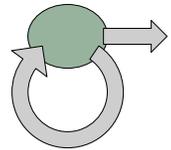
- From product flow

- 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



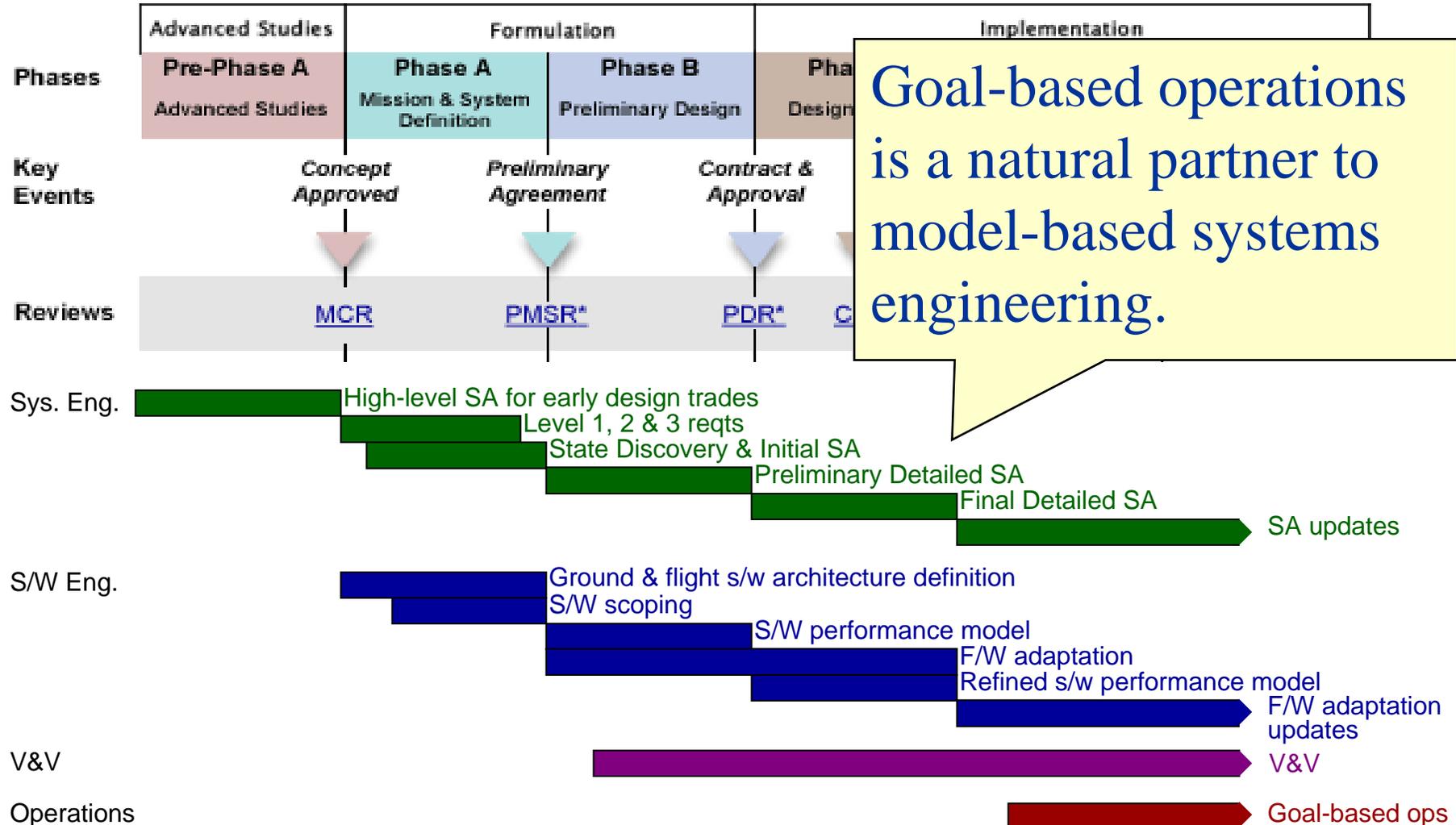
- To work flow

- 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





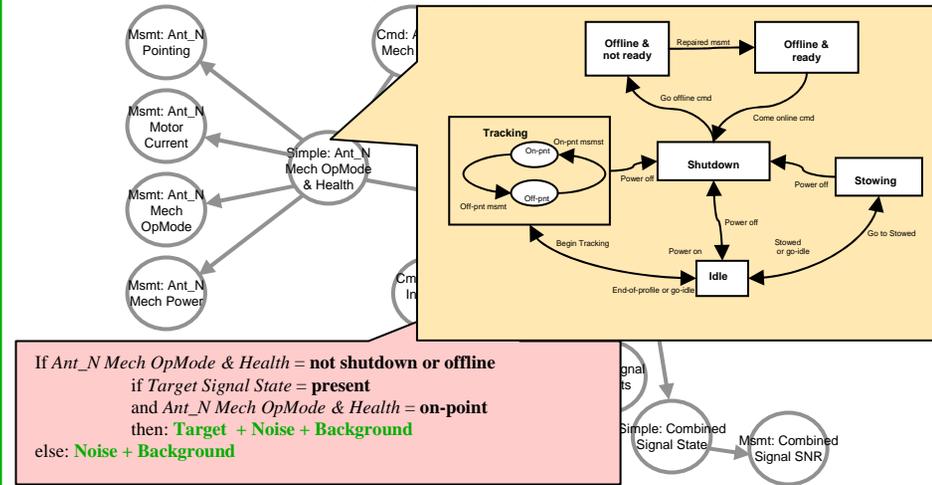
# Integration of goal-based ops into the mission lifecycle



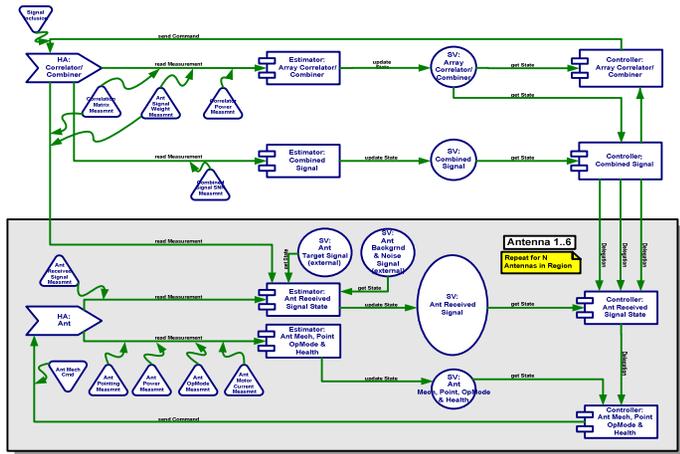
## 1. System to be controlled



## 2. State Analysis produces model

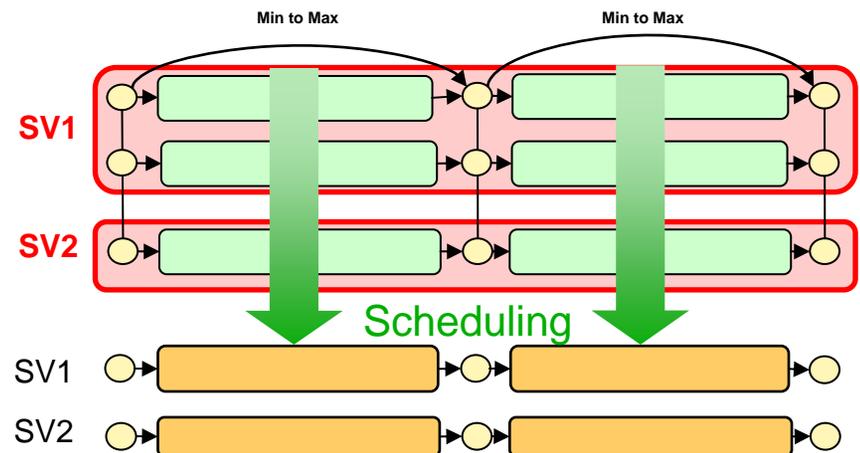


## 3. Model informs software design



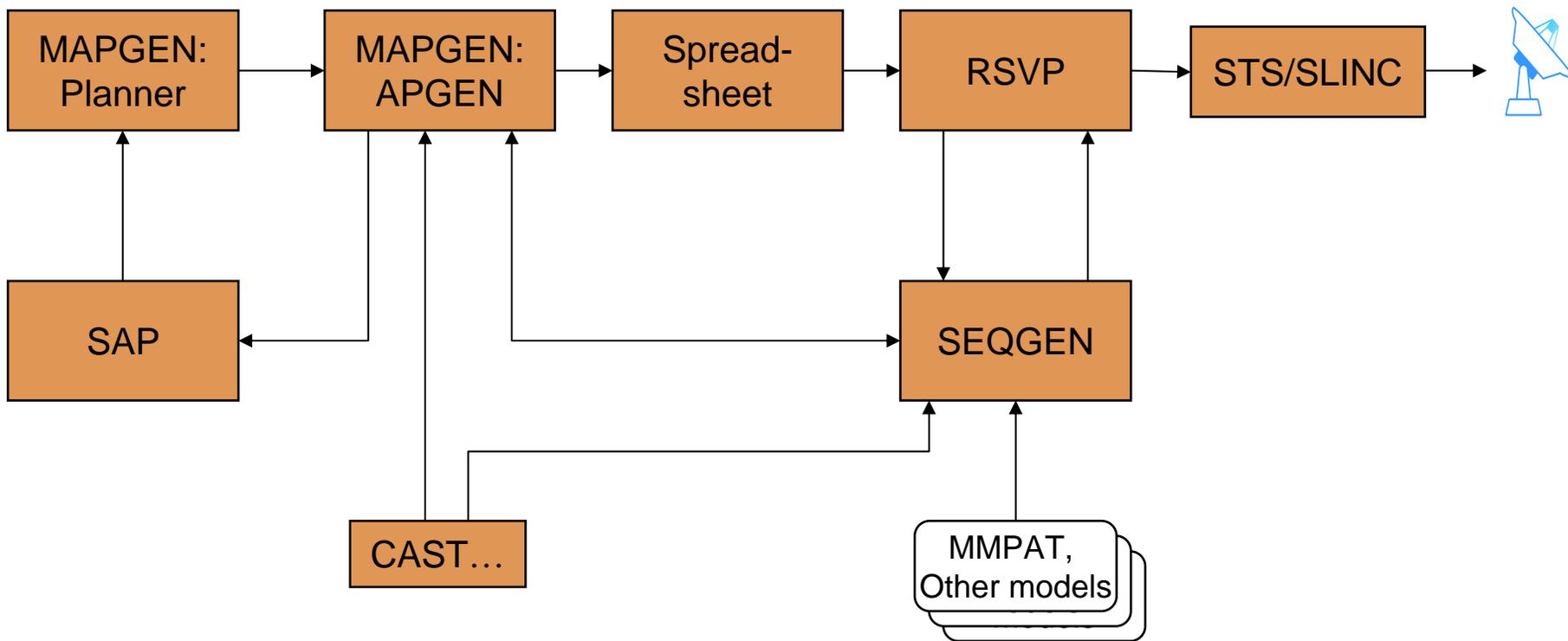
DSAN Collaboration Diagram

## 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

JPL

- 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...



**JPL**

# 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?



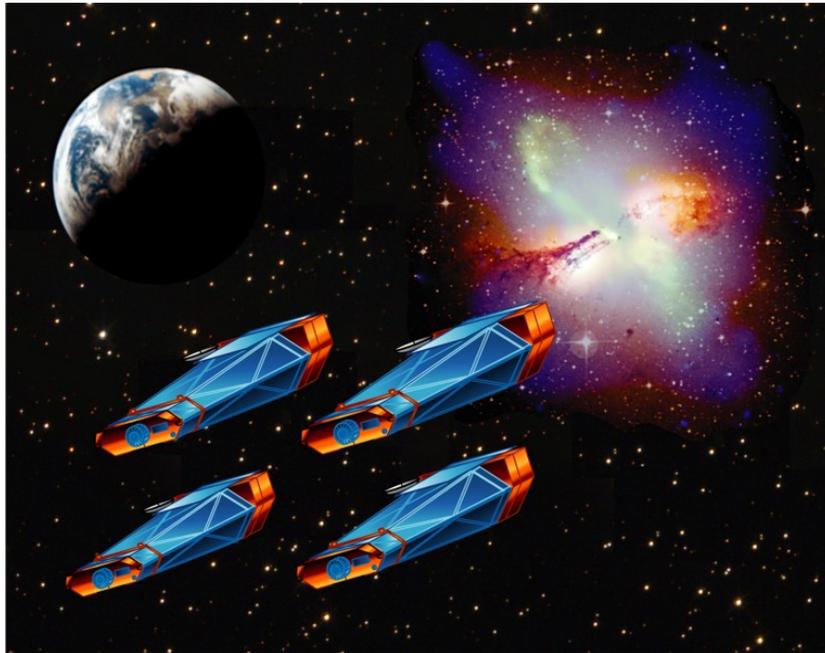
- Current ops approach doesn't really have a “standard”, does it?
  - Each space-faring organization has its own accepted command language and set of ops processes



# Because...

**JPL**

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

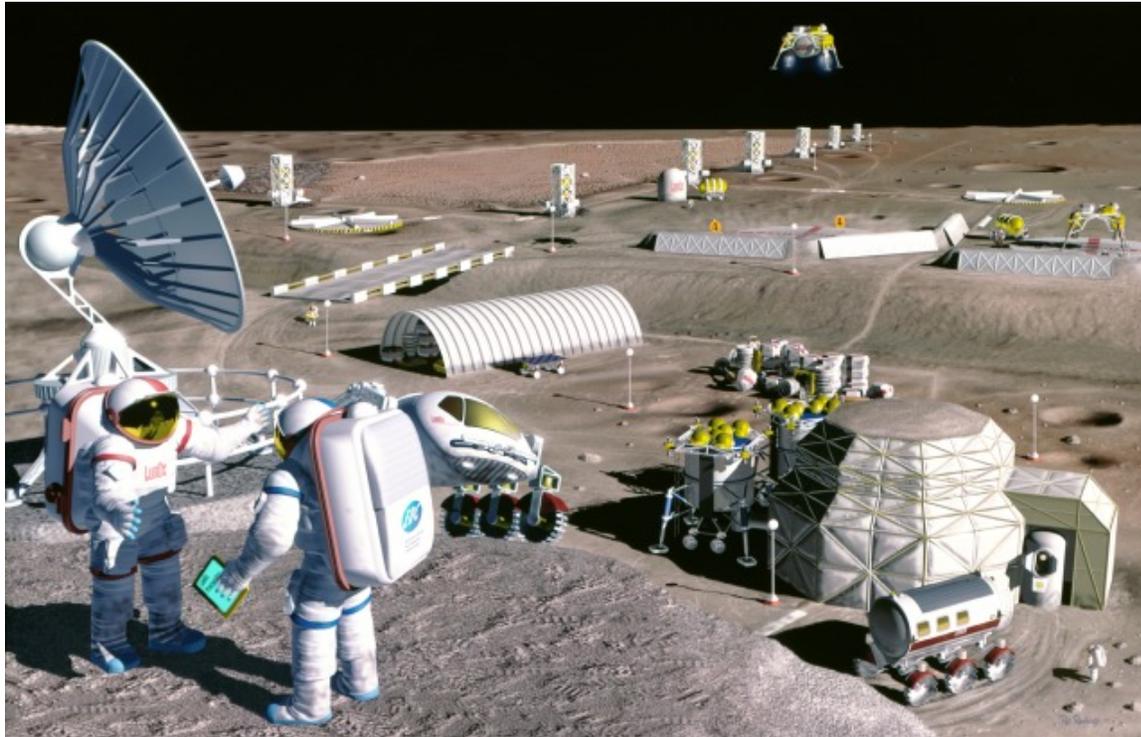




# Because...

**JPL**

- 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...

JPL

- 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?

**JPL**

- 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?

**JPL**

- 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?