10th Anniversary GSAW 06 1997 to 2006
Ground System Architectures Workshop

Working Group Session 10A:

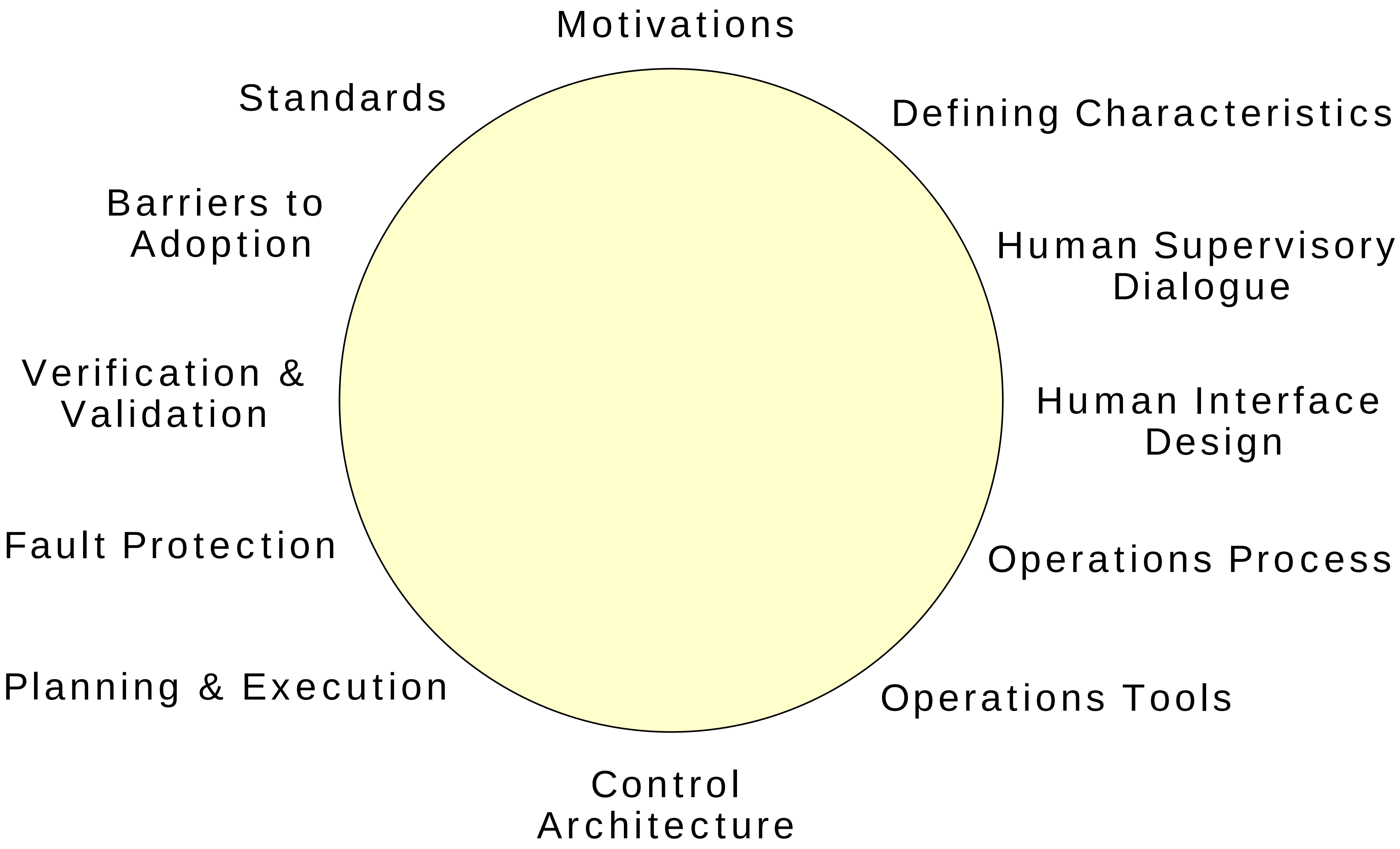
Toward Standards for Goal-Based Operation

Chair:	**Daniel Dvorak,** Jet Propulsion Laboratory, Caltech
Panelists:	**John Gersh,** Applied Physics Lab, Johns Hopkins Univ.
	Mitch Ingham, Jet Propulsion Laboratory, Caltech
	Andrew Rowland, The Aerospace Corporation
	Bonnie Triezenberg, Boeing

What’s the Objective of this Working Group?

- Build a community of interest in goal-based ops
- Raise awareness of motivations and benefits
- Identify issues and set an agenda for a standards effort

Topics of Interest



Motivations

Defining Characteristics

Human Supervisory Dialogue

Human Interface Design

Operations Process

Operations Tools

Control Architecture

Planning & Execution

Fault Protection

Verification & Validation

Barriers to Adoption

Standards

Panel

Daniel Dvorak (chair)

- Principal Engineer: Planning & Execution Systems
  Jet Propulsion Laboratory, California Institute of Technology

John Gersh

- Principal Engineer: Human-Computer Interaction,
  System and Information Sciences Group
  Applied Physics Laboratory, The Johns Hopkins University

Mitch Ingham

- Senior Engineer: Flight Software Systems Engineering & Architectures
  Jet Propulsion Laboratory, California Institute of Technology

Andrew Rowland

- Project Engineer, WGS Mission Integration
  The Aerospace Corporation

Bonnie Triezenberg

- Software Chief Engineer
  Boeing Satellite Development Center

Agenda

1:00	**Overview of goal-based operation**
1:15	**Panel Discussion + Questions:** **Important issues in goal-based operation**
2:15	**Open Discussion**
3:00	**Break**
3:15	**Quick Summary for new attendees**
3:20	**Panel Discussion + Questions:** **What should be in a standard, and why?**
4:15	**Open Discussion**
4:45	**Next Steps**
5:00	**End**

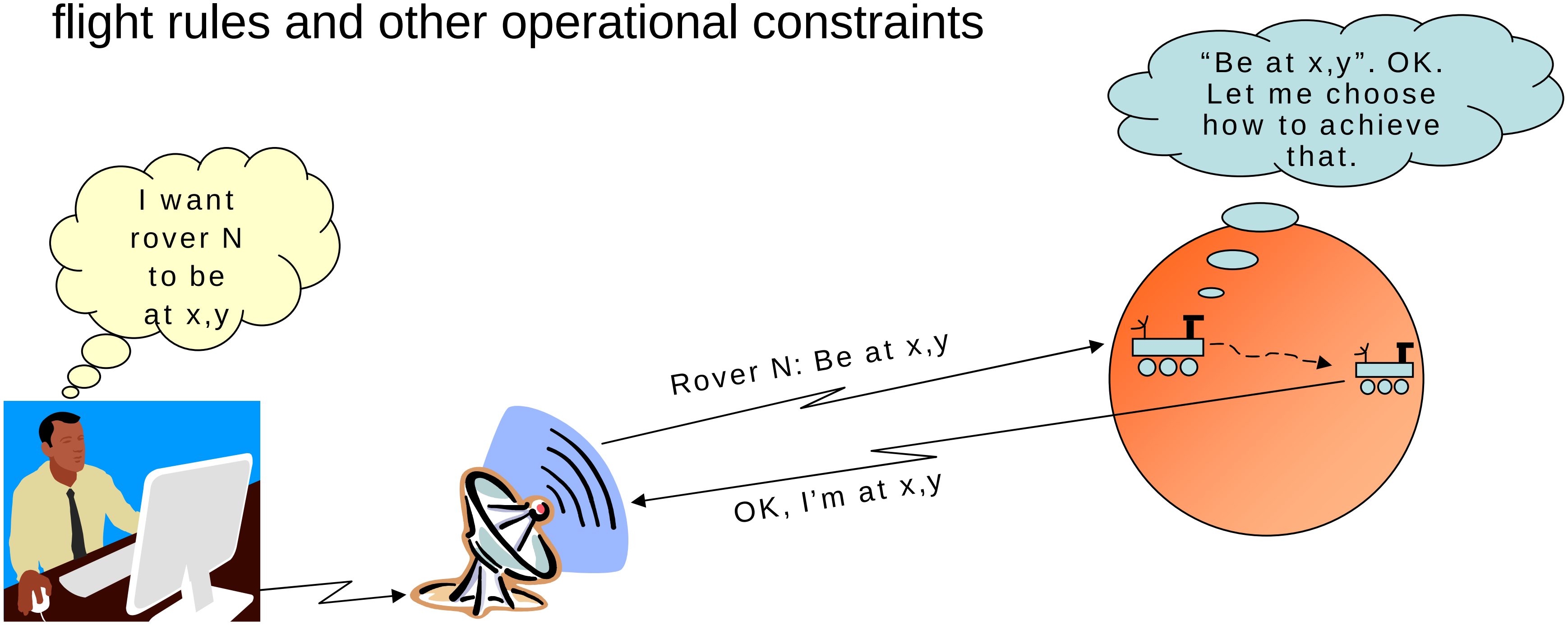
Goal-Based Operation

Overview and Motivations

What is Goal-Based Operation?

A “goal” is an explicit expression of operator or customer <u>intent</u>

- Specify *what* to accomplish, not *how* to accomplish it
- Express intent in an explicitly verifiable form
- Carry expression of intent into the uplink products
- Allow system to select among alternatives to achieve goals
- Intent includes not only activity objectives but also flight rules and other operational constraints


I want rover N to be at x,y
Rover N: Be at x,y
OK, I’m at x,y
“Be at x,y”. OK. Let me choose how to achieve that.

Why Should You Care?

Flexibility, reliability, and robustness

- Systems have a much better chance of …
  - preserving planned functionality, because they know what was *intended* by the original plan
  - responding to opportunities, because they can quickly implement intent *according to local conditions*
- Checking plans becomes more rigorous and complete
- Execution directly monitors results, enabling local fault responses

Operability

- Enables more concurrent, iterative operations planning

Inspectability

- More readable *and* verifiable than sequences, sequence generators, and rule bases

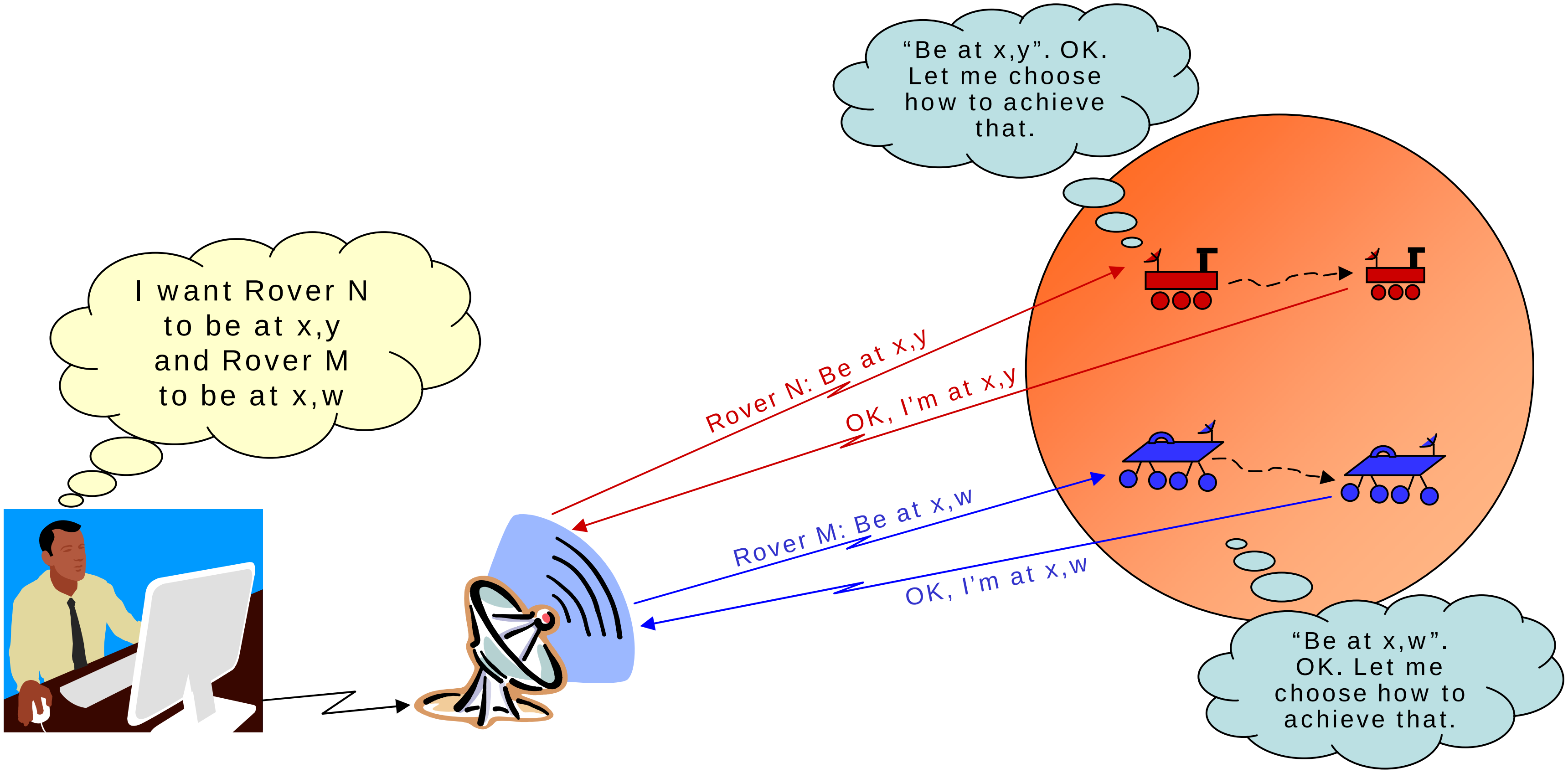
Automation

- Goals are amenable to automated reasoning using domain models
- Easier to encode domain models than rules

Why Do We Need Standards?

Interoperability and reusability

- Goals enable an interoperability standard for control
- Same *high-level* goals can be used by diverse elements of a system
- Hierarchical nature of goals makes them more reusable


"Be at x,y". OK. Let me choose how to achieve that.
I want Rover N to be at x,y and Rover M to be at x,w
Rover N: Be at x,y
OK, I'm at x,y
Rover M: Be at x,w
OK, I'm at x,w
"Be at x,w". OK. Let me choose how to achieve that.

Goals: Some Defining Characteristics (1 of 2)

- **A goal specifies an objective to be accomplished (operator or customer intent)**
  - A goal specifies a desired state, not the commands necessary to achieve it
  - It leaves options for a control system in how to achieve the goal based on local knowledge
  - It is explicit, compact, and inspectable
  - The activity to achieve the objective plays out over time

- **A goal can be low-level or high-level**
  - “switch 12 is closed from 2pm to 3pm”
  - "camera boresight is pointed within 1 milliradian of Polaris from 2pm to 3pm"

Goals: Some Defining Characteristics (2 of 2)

- **During execution a goal either succeeds or fails (principle of cognizant failure)**
  - Execution is monitored
  - Nominal execution and fault reactions both use goals
  - Facilitates verification (self-checking)
  - Goals imply closed-loop control
- **Multiple goals can be coordinated to achieve complex objectives**
  - Hierarchical expansion (a goal can have subgoals)
  - Ordering and timing dependencies
- **Conflicts among goals can be detected and avoided during scheduling**

Motivations for Goal-Based Operation

- Reduce operations costs
  - Easier to specify *what* to do than *how* to do it
  - Plans are more compact and inspectable
  - Detailed steps generated automatically

- Reduce operations risks
  - Systems are too complex for operators to know, in all cases, how to transition from state A to state B

- More effective use of operators
  - Automation frees operators to focus more on the big picture (what humans do best)

- More effective use of assets
  - Onboard closed-loop control enables full use of capabilities in the face of intermittent communications and long light time delays
  - Detect short-lived science events
  - Military reconnaissance events

- Operation of diverse elements
  - Goals provide a *lingua franca* for operation of system elements from multiple suppliers

- Integrated fault protection
  - Goal-based control architecture enables integral fault protection
  - Avoids awkward, hard-to-engineer relationship between sequencer and fault protection

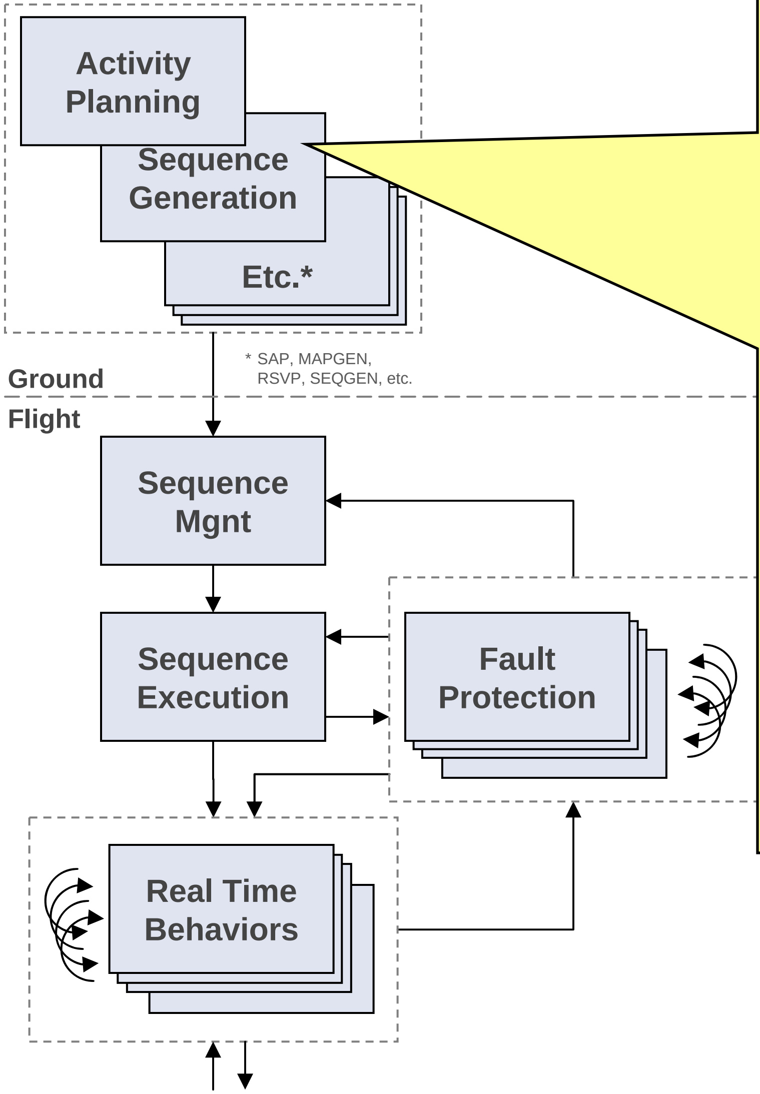
- More robust systems
  - In situ decision-making enables quick reaction to events
  - System can react to hazards and faults

- System design simplification
  - Goals can represent several forms of intent: activities to be achieved, flight rules, events to pursue opportunistically

- Verification & Validation
  - Goal-based execution is inherently self-checking

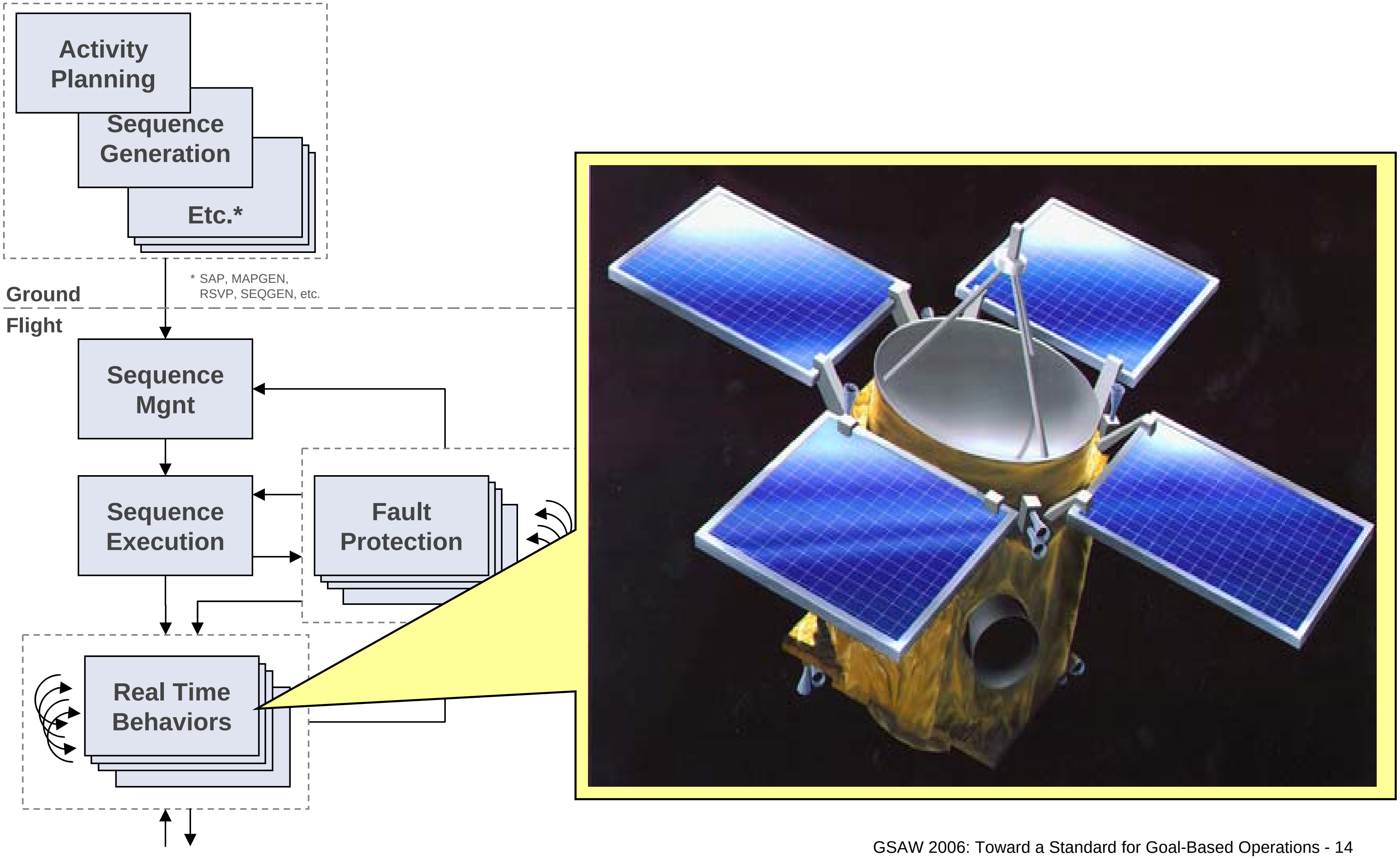
- Leverage increased computing power on flight systems

Current State of the Practice



* SAP, MAPGEN, RSVP, SEQGEN, etc.

Current State of the Practice



Activity Planning

Sequence Generation

Etc.*

* SAP, MAPGEN, RSVP, SEQGEN, etc.

Ground

Flight

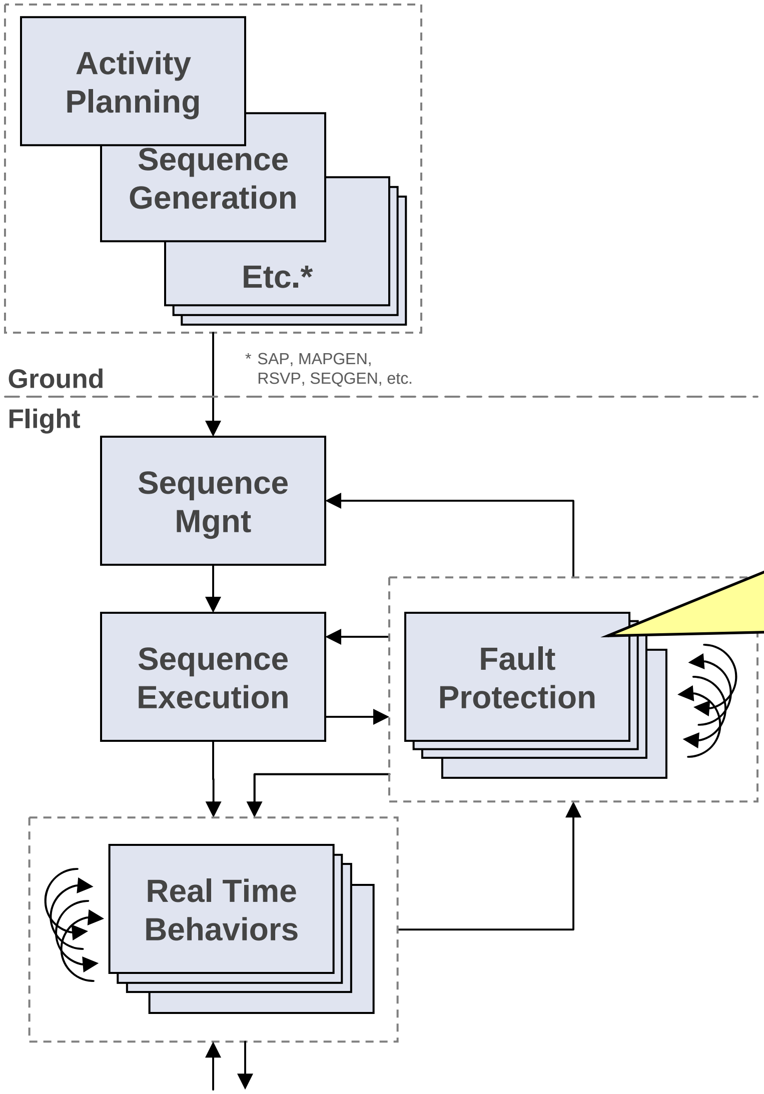
Sequence Mgnt

Sequence Execution

Fault Protection

Real Time Behaviors

Current State of the Practice



Activity Planning

Sequence Generation

Etc.*

* SAP, MAPGEN, RSVP, SEQGEN, etc.

Ground

Flight

Sequence Mgnt

Sequence Execution

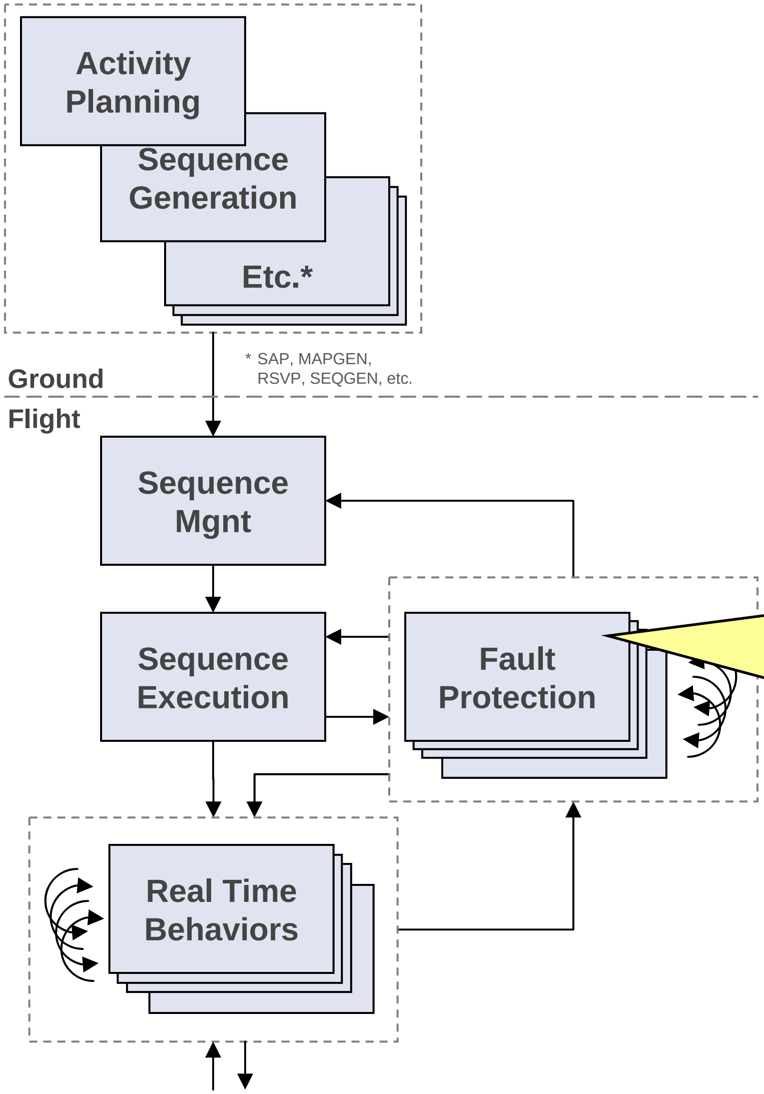
Fault Protection

Real Time Behaviors

Fault protection software running in parallel, ready to “take over” from nominal sequence execution when a fault monitor is triggered. The usual off-nominal response is “safe mode”:

- **costly ground ops**
- **lost science opportunities**

Current State of the Practice



Activity Planning

Sequence Generation

Etc.*

* SAP, MAPGEN, RSVP, SEQGEN, etc.

Ground

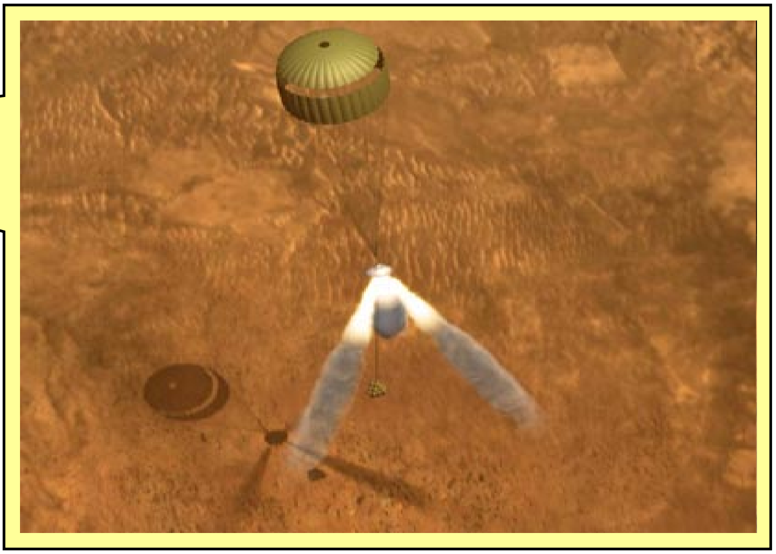
Flight

Sequence Mgnt

Sequence Execution

Fault Protection

Real Time Behaviors



Commands vs. Goals

All commands direct momentary *changes* of state, …

- **But many commands are open-loop**
  - **Examples: open a valve; select an antenna; set a mode…**

- **Typically depend only on intrinsic state stability**
  - **Persistence of effects is assumed, not enforced**
  - **Failure to effect or sustain a change may go unnoticed until subsequent dangers trigger a fault response**

Commands vs. Goals

Goals, a.k.a. closed-loop commands, change *objectives* on state

- **Common in most space systems, but not the norm**
  - **Examples: Track the earth; take a picture; drill a hole…**
- **Subsequent action monitors and sustains the objective**
  - **Playing out over time is a defining characteristic**
  - **Failure to achieve an objective is overt and recognized early**
- **More general representation**
  - **A goal can mimic *any* open-loop command**
  - **No hidden assumptions, so easier to construct, schedule, and verify robust sequences**
- **Goals can also specify passively achieved behavior**
  - **Flight rules and constraints, resource management, fault monitoring can use same representation as nominal “sequence”**