Goal-based Operations: Leveraging Automated Planning for Space Exploration

Michel D. Ingham, Sc.D.
Senior Software Systems Engineer
NASA Jet Propulsion Laboratory

AIAA Infotech@Aerospace Conference
Intelligent Systems in Aerospace Workshop
Panel I – Strategic Planning for Space and Planetary Robotic Missions
Arlington, Virginia
September 26, 2005
Outline

- State of the Practice
- Goal-based Operations
- Steps in the Right Direction
- Reaping the Benefits
- Challenges
Current State of the Practice

Activity Planning

Sequence Generation

Etc.*

Ground Flight

Sequence Mgmt

Sequence Execution

Fault Protection

Real Time Behaviors

* SAP, MAPGEN, RSVP, SEQGEN, etc.
Current State of the Practice

- Activity Planning
- Sequence Generation
- Fault Protection
- Etc.

* SAP, MAPGEN, RSVP, SEQGEN, etc.

If absolutely necessary, conditional behavior (event-driven execution) via rule-based monitors or hard-coded state machines.

Real Time Behaviors

Ground
Flight

Sequence Mgmt

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

For critical mission sequences, standard safing mechanism is disabled - hard-coded fault protection provided by highly-specialized s/w modules:

- ad-hoc
- complex
- expensive to generate and test

*SAP, MAPGEN, RSVP, SEQGEN, etc.
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”
Goal-based Operations

• Say WHAT to do, not HOW to do it
  – Operator’s intent is explicit
  – More compact and inspectable
  – Easier to see interactions and conflicts between activities

• Allows for both time- and event-driven execution

• Allows for hierarchical expansion

• Bottom-line motivation:
  – Reduce ops costs (decrease comm bandwidth needed for control, enable use of onboard autonomy)
  – Reduce risk (facilitate integral fault protection)
Steps in the Right Direction (1)

Deep Space One

Goals
- Mission Manager
- Scripted Executive
- HSTS: Planner/Scheduler
- Livingstone: Diagnosis & Repair

Scripts
- Planning models
- Component models
Steps in the Right Direction (2)

MAPGEN: Planner
MAPGEN: APGEN
Spreadsheet
RSVP
STS/SLINC
SAP
CAST...

Developed by NASA ARC & JPL;
MER Ops personnel use MAPGEN to:
- Plan Goals
- Analyze Resources
- Edit Plans
Reaping the Benefits: Robustness

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

JPL’s Mission Data System
Reaping the Benefits: Robustness

- Control layer has flexibility in achieving goal
- Enables integration of tiered fault management capabilities
- Enables integration of state-of-the-art autonomy software

MIT’s Titan Model-based Executive
Reaping the Benefits: Greater Science Return

Autonomous Sciencecraft Experiment on EO-1

Image taken by Spacecraft (hyperion) & appropriate bands extracted

Cloud Detection

Feature Detection

Clouds Sparse

Extensive Cloud Cover

No feature Detected

Feature Detected

Downlink Image

Onboard Replanning

Retarget for New Observation Goals

New Science Images
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 planning & scheduling (V&V of goal-based planning & scheduling capabilities)?
• How do we overcome the “cultural” hurdles to acceptance of these new methods and tools?
Backup Slides
(Details on each Challenge Question)
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

Our goal:

✓ head toward unified representation of spacecraft
✓ accommodate complexities of spacecraft domain
✓ maintain capacity for knowledge abstraction
Transitioning from “product flow” to “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

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

3. Model informs software design

4. Model informs operations

If $\text{Ant}_N$ Mech OpMode & Health = not shutdown or offline
if Target Signal State = present
and $\text{Ant}_N$ Mech OpMode & Health = on-point
then: Target + Noise + Background
else: Noise + Background

SV1

SV2

Scheduling
Adapting legacy tools

MAPGEN: Planner

MAPGEN: APGEN

Spreadsheet

RSVP

STS/SLINC

SAP

CAST...

SEQGEN

MMPAT, Other models
V&V of goal-based planning & scheduling 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., NMP)
  - 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 issue…