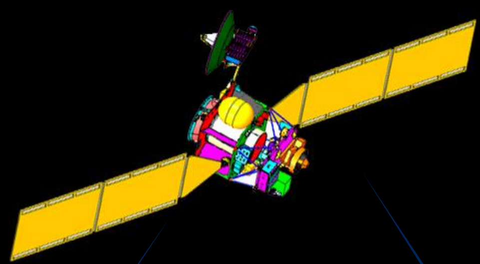
Discussion of Intelligent Data Understanding, Autonomous Spacecraft Operations and On-board Computing

(TA11)

David P. Watson
dave.watson@jhuapl.edu
(240) 228-5902

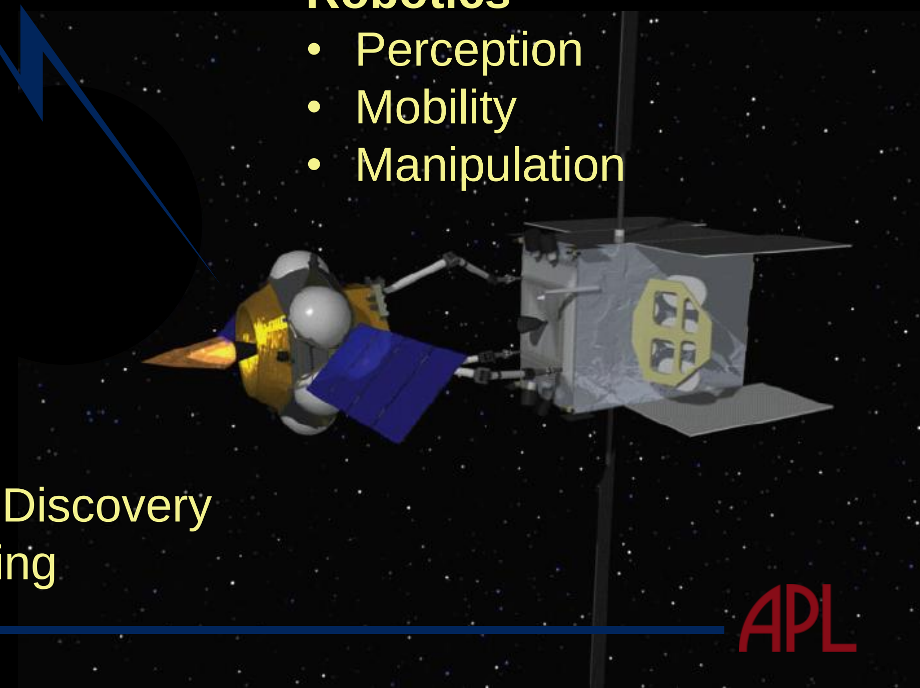
The Johns Hopkins University
APPLIED PHYSICS LABORATORY

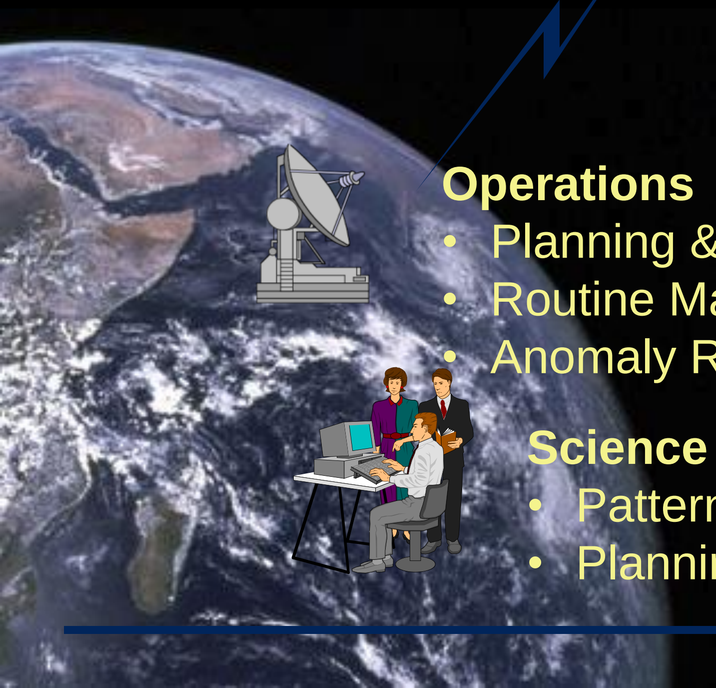
Space Applications of Autonomy



Flight Systems
- Opportunistic Sensing
- Fault Management
- Event Response
- Coordination

Robotics
- Perception
- Mobility
- Manipulation



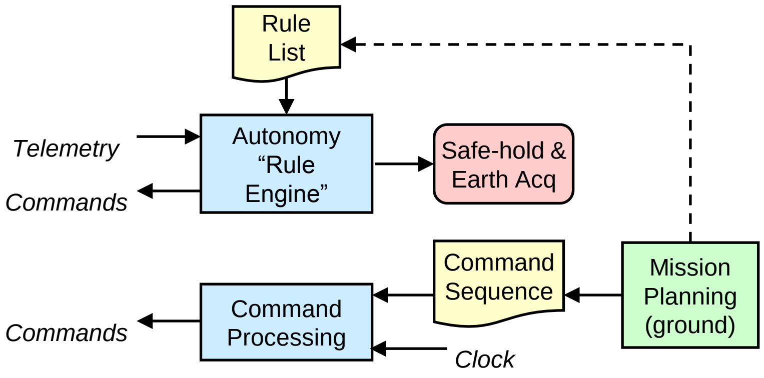


Operations
- Planning & Scheduling
- Routine Maintenance
- Anomaly Resolution

Science
- Pattern & Semantic Discovery
- Planning & Scheduling

APL Autonomy for Deep Space Missions

- **New Horizons (launch 2006; $470M)**
  - Pluto Mission, 9 year transit, 9hr communications latency, 36 hr encounter phase
- **STEREO (launch 2006; $210M)**
  - Solar science mission, twin vehicle deep space operations
- **MESSENGER (launch 2004; $330M)**
  - Mercury mission, 5 year transit, remote orbital operations



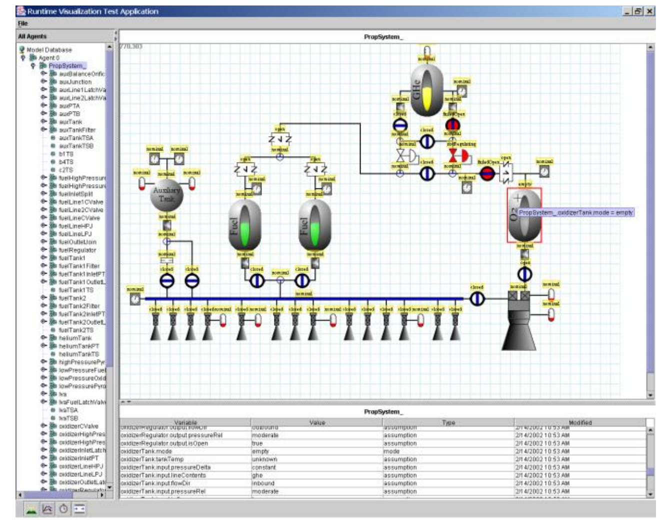
Flight Rule Sets:

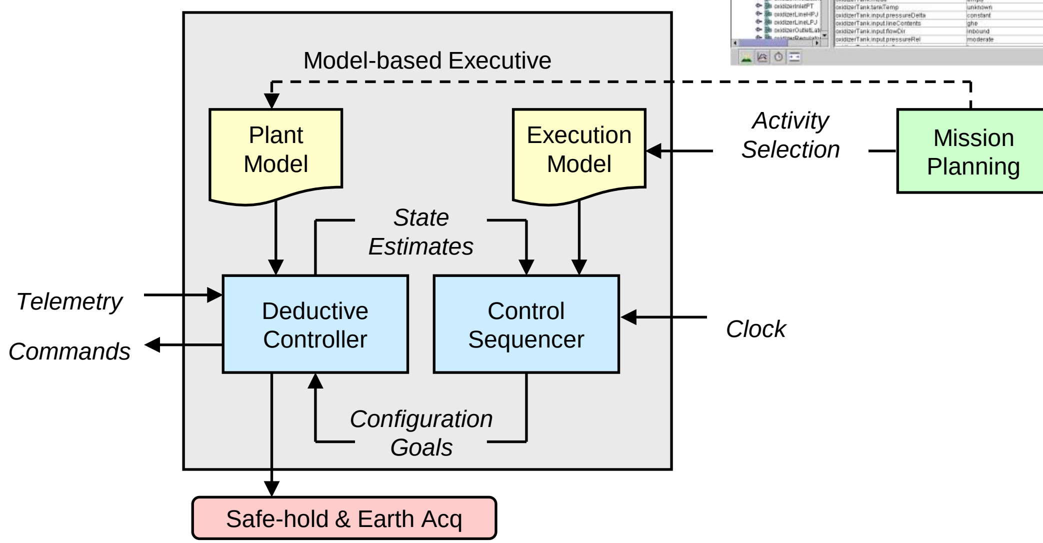
- NEAR: 16
- STEREO: 140
- MESSENGER: 240
- New Horizons: 118

New Horizons Autonomy Testing

- 8 staff, 10 months
- Rule set mod for encounter phase

Model-based Autonomy R&D





Flying Advanced Autonomy

- **Deep Space 1 Remote Agent Experiment (1999)**
- **EO-1 Autonomous Sciencecraft Experiment (2003)**
- **Other Proposed Efforts**
  - New Millennium ST7 Mission
  - Mission Data System (JPL)
  - MESSENGER Autonomy Experiment

Principles of FM Autonomy

- **Understandability:**

Understandability defines the ability to design, display and review the autonomy system such that non-software domain experts or system engineers can understand the design.

  - **Necessary for reviews:** FM is multi-disciplinary and need all subsystems understanding the ConOps to produce good designs
  - **Essential for future modifications:** Better context is key to making the right change and translating need into implementation

- **Flexibility:**

Flexibility defines an ability to modify the design pre- and post-launch in parts without patching or complete code uploads.

  - **Speeds development and testing:** Decouples autonomy from FM; enables testing outside of nominal flight bounds.
  - **Eases burden on operations staff:** Situations demanding workarounds can be performed though on-board changes; ensures ability to go lights-out.

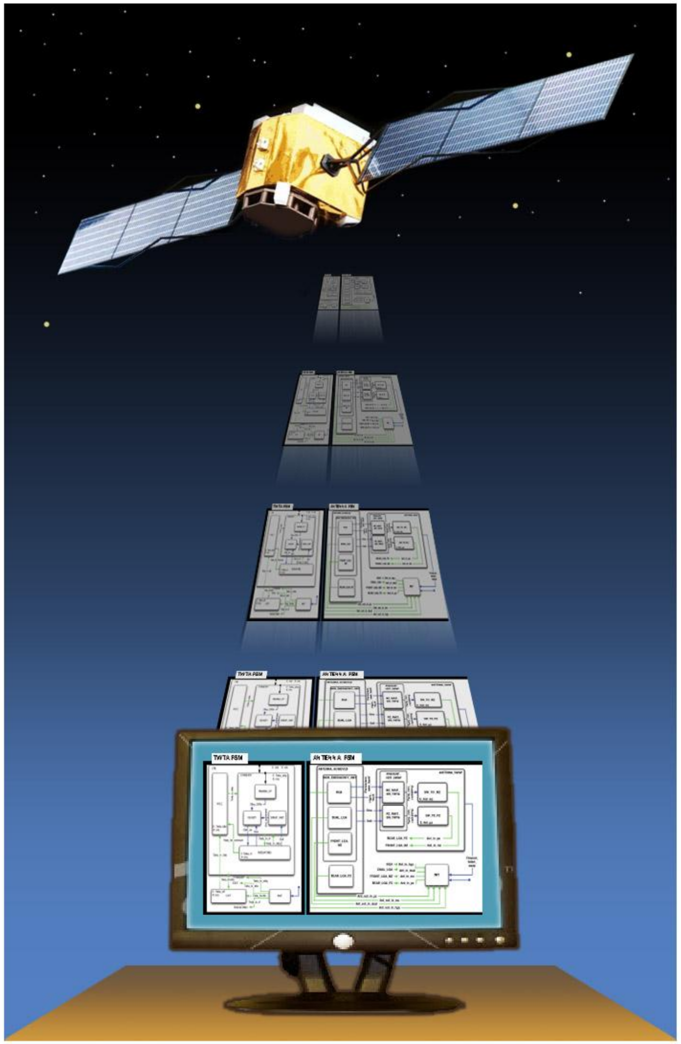
- **Verifiability:**

Verifiability defines the ability to exhaustively and rapidly verify the autonomy system.

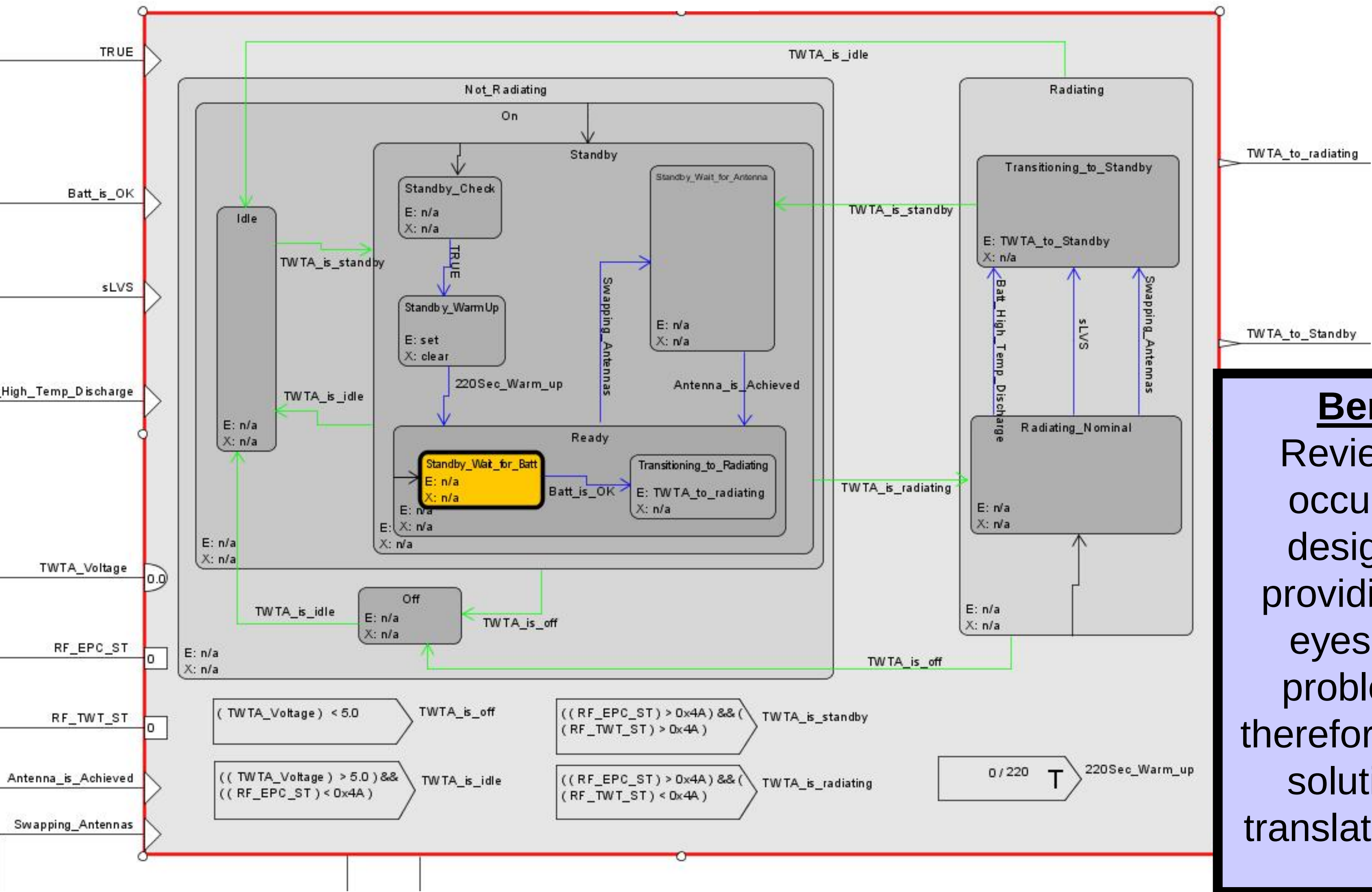
  - **Prevent crunch in I&T testing:** Provides early on testing
  - **Ensure risk level:** Current testing may not find or see all problems

Uploadable Executable Specification Autonomy (ExecSpec)

- Domain Experts or system engineers draw state diagrams to represent desired behavior using interactive development environment (IDE)
- Design can be easily reviewed
- User-driven or user-scripted simulation
- Automatic Verification (NuSMV) based on project requirements
- Diagrams uploaded into the spacecraft (no code)
- On-board diagram interpreter
- Design can be further modified in real-time at any time pre- or post-launch (no patching or recompiling)
- Autonomy is visualized during test or flight by animating diagrams (same consistent interface from design to test to operate)



Understandability with ExecSpec Example (RF Amplifier Diagram)



Benefit:
Reviews can occur at the design level providing more eyes on the problem and therefore a better solution. No translation errors

Spacecraft Autonomy: Some Thoughts

- Current state of practice will not scale to meet next generation challenges. Step change is needed.
- Autonomy can be viewed as an extension of Fault Management and G&C.
- Next generation autonomy will require significant cultural changes across the full spectrum of spacecraft systems engineering.
- There are, perhaps, lessons to be learned from commercial industry (e.g. automobile, SCADA).
- Verification and Validation is a significant issue.

TA11 Roadmap Comments

- **Top Technical Challenges**
  - Verification and Validation of Autonomous & Adaptive Systems
- **Technology Gaps**
  - Computing hardware between multi-core and quantum (e.g. analog pattern classification)
  - Case-based emphasis to parallel Model-based direction
  - Hybrid Discrete/Continuous model-based systems
- **High Priority Areas**
  - Languages, Tools, Training, and Testbeds for model-based programming
- **Alignment with NASA expertise & role**
  - Good alignment across multiple centers

Comments (cont.)

- **Competitive Placement**
  - Automotive Industry?
  - SCADA Industry?
- **Game-changing Technology**
  - Formal/Automated Software Validation
- **Technology Near Tipping Point**
  - Executable Specification
- **Time Horizon**
  - 5 to 10 year time horizon for adoption
- **Value/Risk**
  - Near Earth: Cost Reduction
  - Deep Space: Risk Reduction, Mission-enabling

Additional Comments/Ideas

- **Significant overlap with TA04**
  - Mirrors "robotics vs. autonomy" dichotomy in the community
- **Multi-user Virtual Environments (MUVE) not mentioned in the collaborative work discussion.**
- **Machine Learning is mentioned, but not emphasized.**
  - Possible role in model-based systems
- **Need a taxonomy of models and a plan for "translation" or adaptation of models across engineering domains.**
- **Really need to recognize the cultural/training challenges associated with model-based design & development.**
  - SysML probably won't be enough

JOHNS HOPKINS
UNIVERSITY

Applied Physics Laboratory