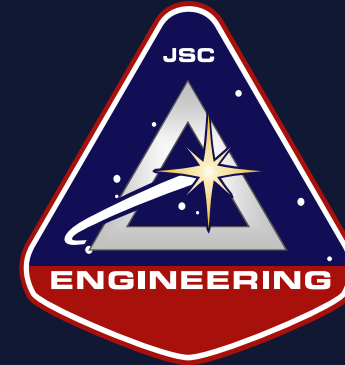


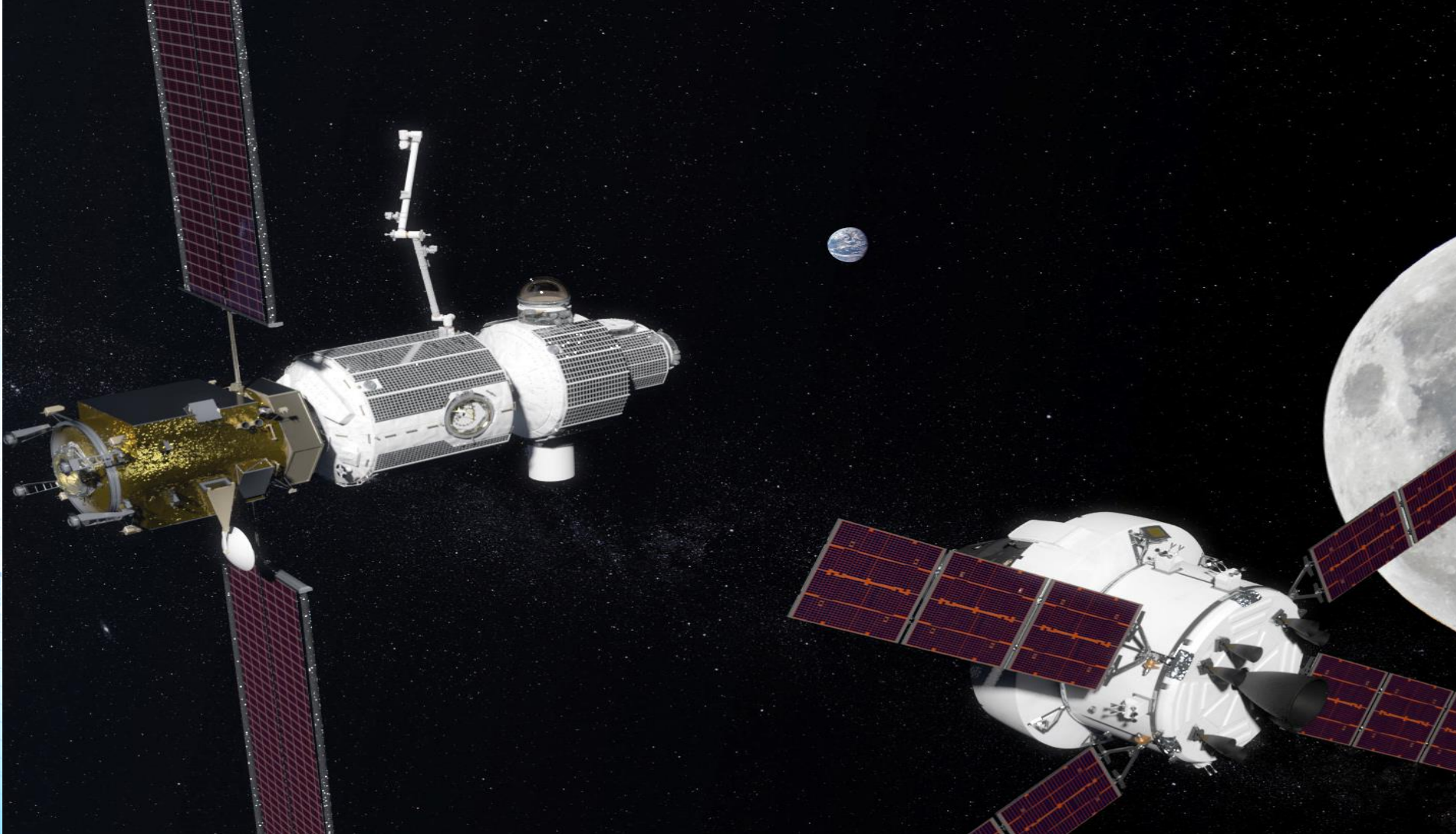


Robotics and Autonomy: The Future of People in Space

Julia Badger, PhD
NASA-Johnson Space Center



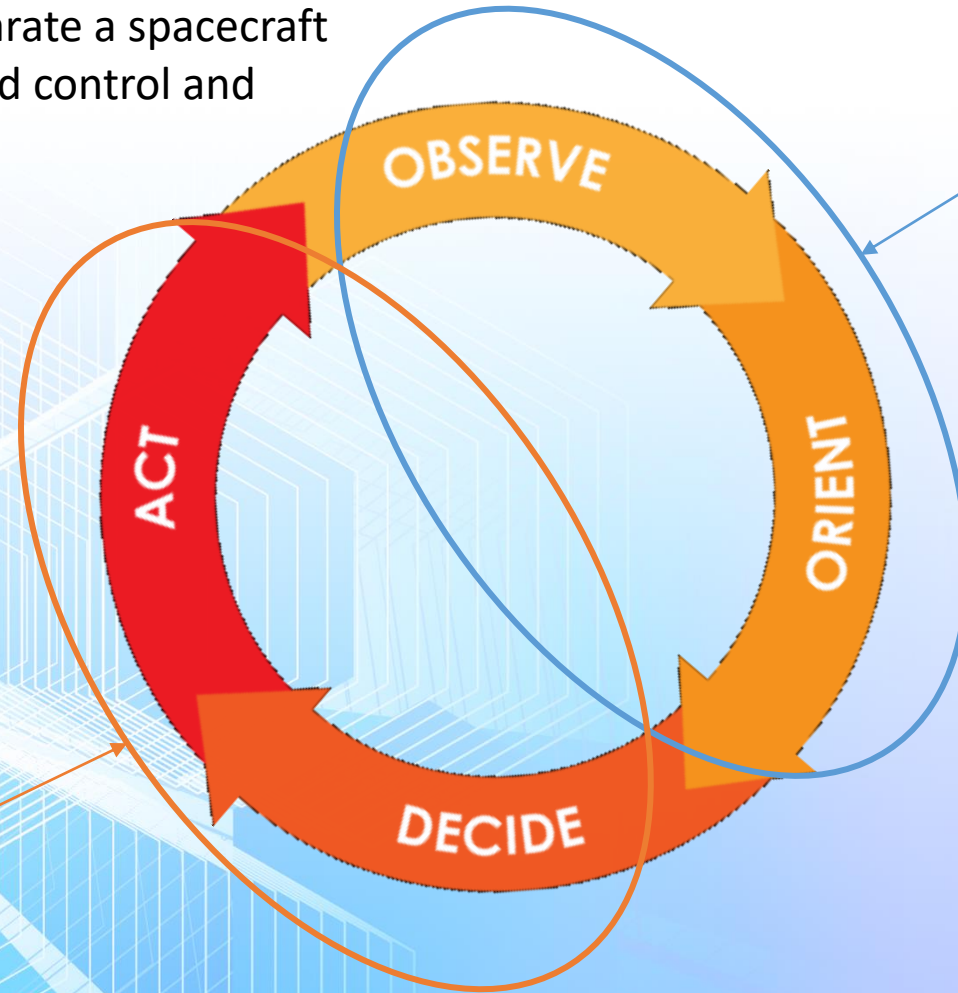
Future Exploration Missions



What is Autonomy?



Autonomy is the ability to separate a spacecraft (and its crew) from Earth-bound control and oversight.



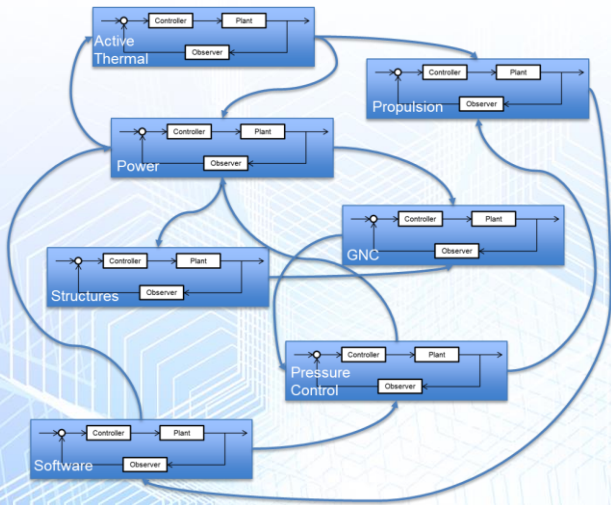
State Analysis- understanding the state of the system
Includes fault detection and isolation.

Plan & Execute- affecting the state of the system
Includes fault recovery.

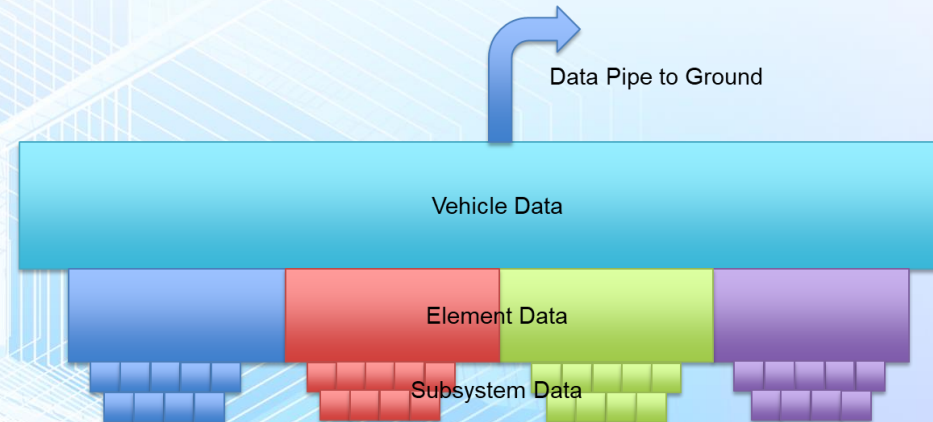
Why Autonomy?



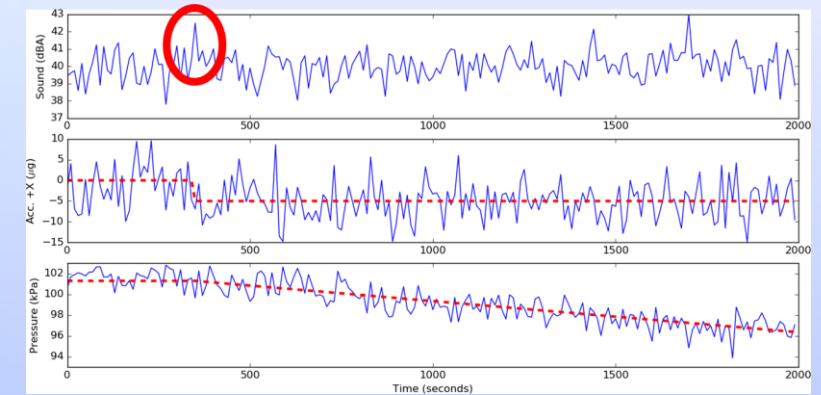
1. Complex system of systems.



2. More data than bandwidth.



3. Critical functions have short time to effect.



Autonomous functions are needed vehicle-wide for both nominal and off-nominal operations.

Autonomy Gaps



- Integrated vehicle systems status, fault response, planning, and control
 - Currently relies heavily on ground control
- Contingency management across many subsystems
 - Particularly leaks and emergencies (failures that currently require hands-on response from crew)
 - Currently relies heavily on both crew and ground control
- Data management and situational awareness
 - Crew commonly provides sensing, sampling, and processing
 - All ISS sensor data is delivered to the ground
 - Ground controllers provides nearly all data analysis

Ways to Achieve Autonomy

- **System design** plays a major role
 - Early definition of subsystem interdependencies is key
 - Simplified interfaces, less complexity, and materials selection for more robust design
 - Robust, fail-operational designs for critical components
 - Make choices to increase the time to criticality
 - Design for robotic maintenance and inspection
- **Vehicle systems management** software provides in situ operational autonomy
 - Distributed, hierarchical architecture
 - ***Clear definitions of interfaces and interdependencies***
 - Careful design of locus of authority
 - Redundancies for data collected in case of failure or degradation

Make system
simple.

Get simple
software.

Robonaut 2 (R2)

- Started in 2007 with GM
 - Leveraged Robonaut 1 technology (1998-2006)
- Common goals
 - Use humans' tools
 - Safely share humans' workspace
 - Do real (useful) work
- Launched on STS-133 in Feb 2011



Robonaut 1, Units A & B



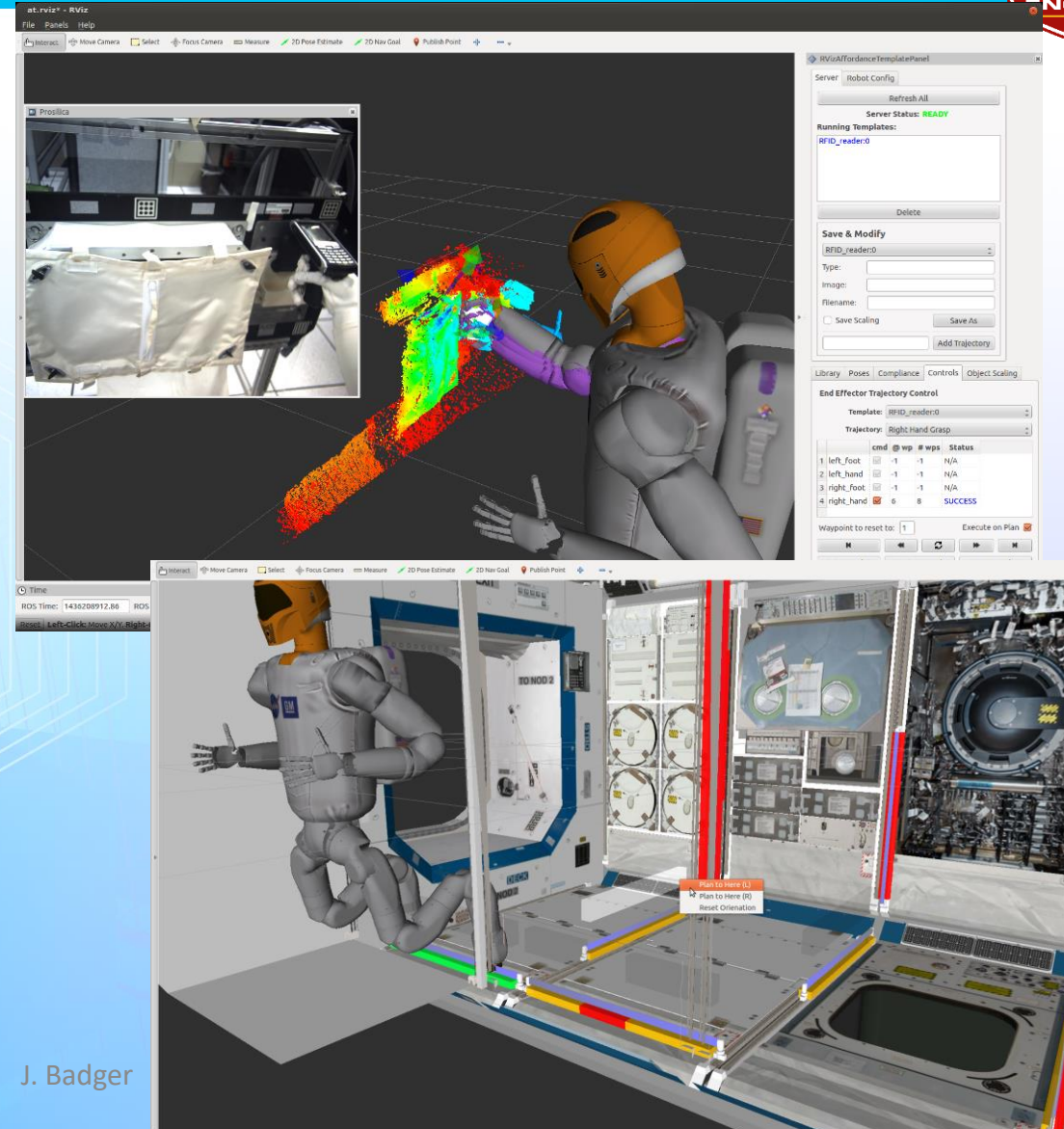
Tasks: Taskboard- Softgoods Panel



Affordance Templates



- Adopted this approach to move from supervised control to autonomous robotic behaviors
- Adapted from concept attempted during first DARPA Robotics Challenge
- Framework upgrades and improvements:
 - Embedded collision data & checking
 - Allowable Collision Matrix
 - Obstacle Avoidance
 - Planner Plugins
 - Customizable planners and trajectory generators
 - Active supervisors
 - QR Code Detection
 - Automatic Object Recognition



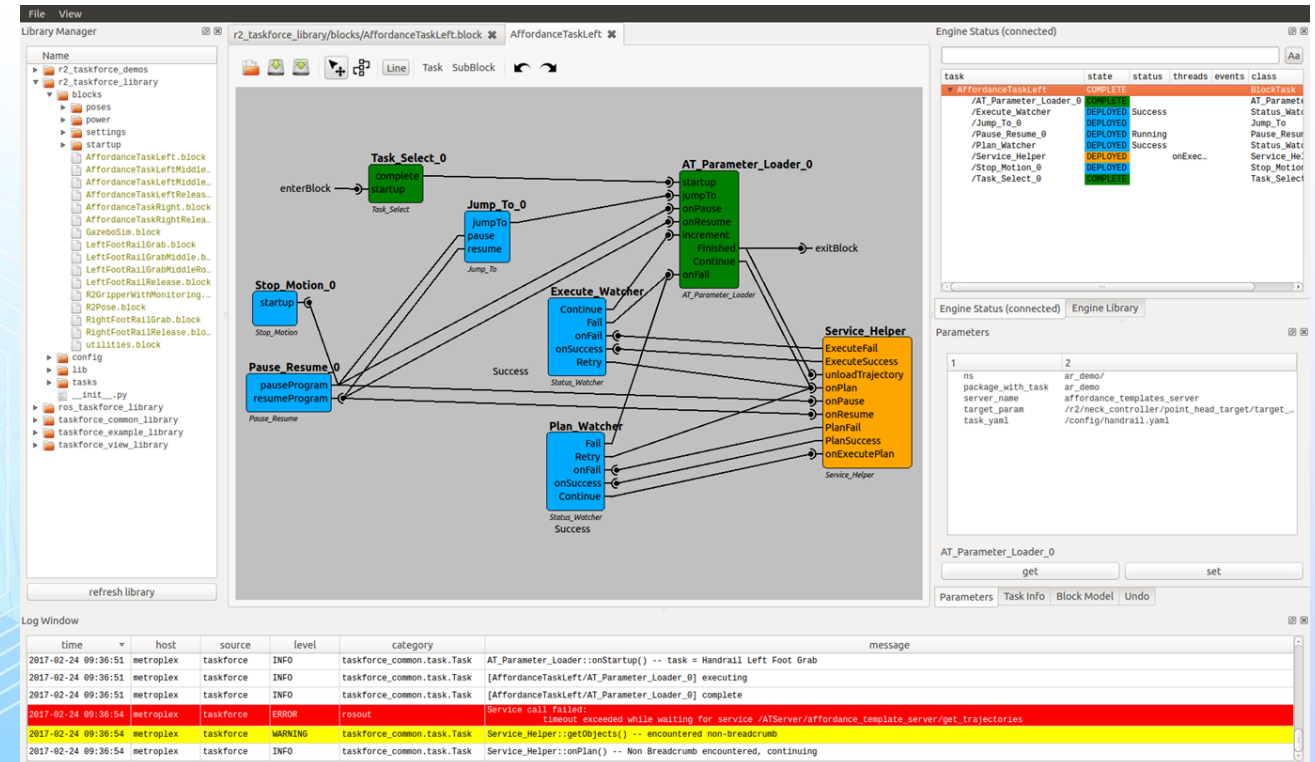
Autonomous Caretaking Demonstration



TaskForce



- General-purpose algorithm design and execution framework that can serve as an Integrated Development Environment (IDE) for complex task development
- Includes options for procedure execution, deployments of task supervisors



Autonomous Logistics Demonstration

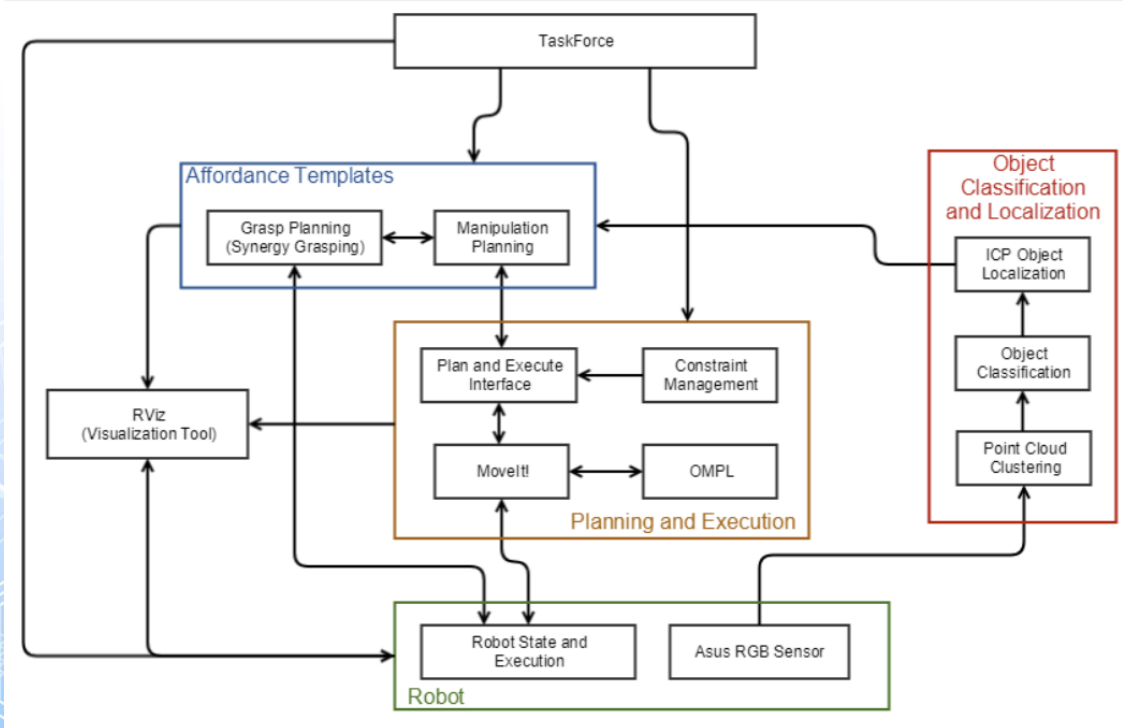


4x

A Software System for Whole-Body Manipulation

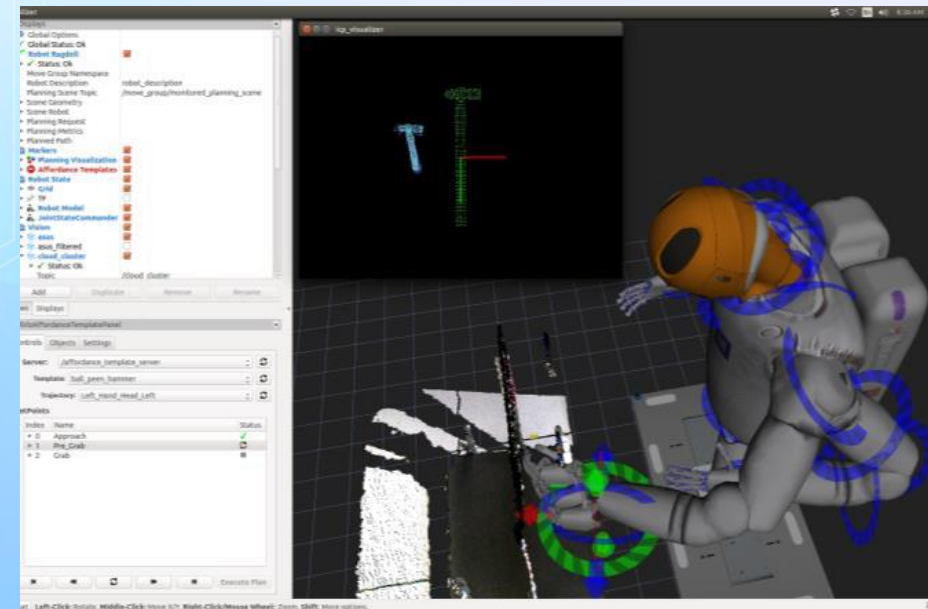
Zachary Kingston, Logan C. Farrell, Michael Park, Mark Moll, Julia Badger, Lydia E. Kavraki

Manipulation Framework



Affordance Templates- framework that uses models of objects encoded with afforded grasps and manipulations registered to the robot's frame of reference to enable tool use.

- Centers around Affordance Template framework and Planning and Execution engine



Cognitive Dexterity

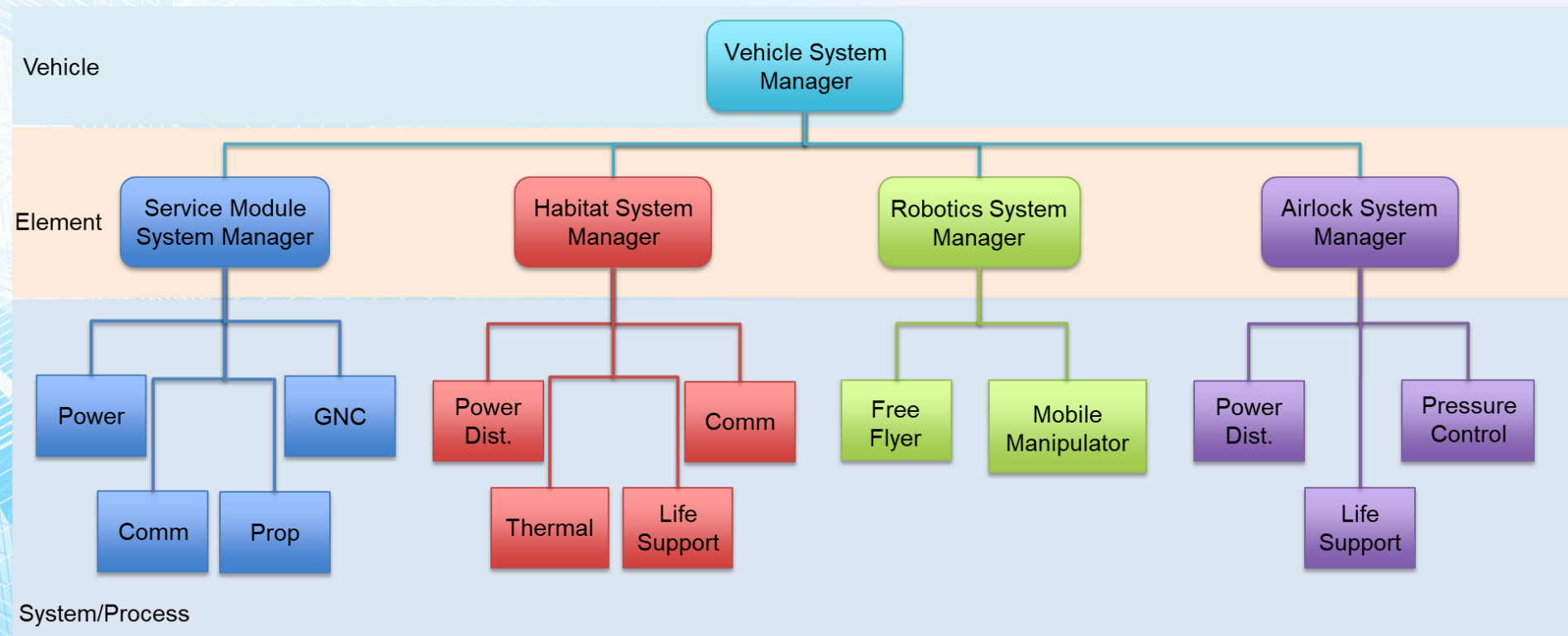


Vehicle System Management



- **Vehicle systems management** software provides in situ operational autonomy
 - Distributed, hierarchical architecture
 - Clear definitions of interfaces and interdependencies
 - Careful design of locus of authority

Conceptual diagram of vehicle following the Autonomous Systems Management Architecture.

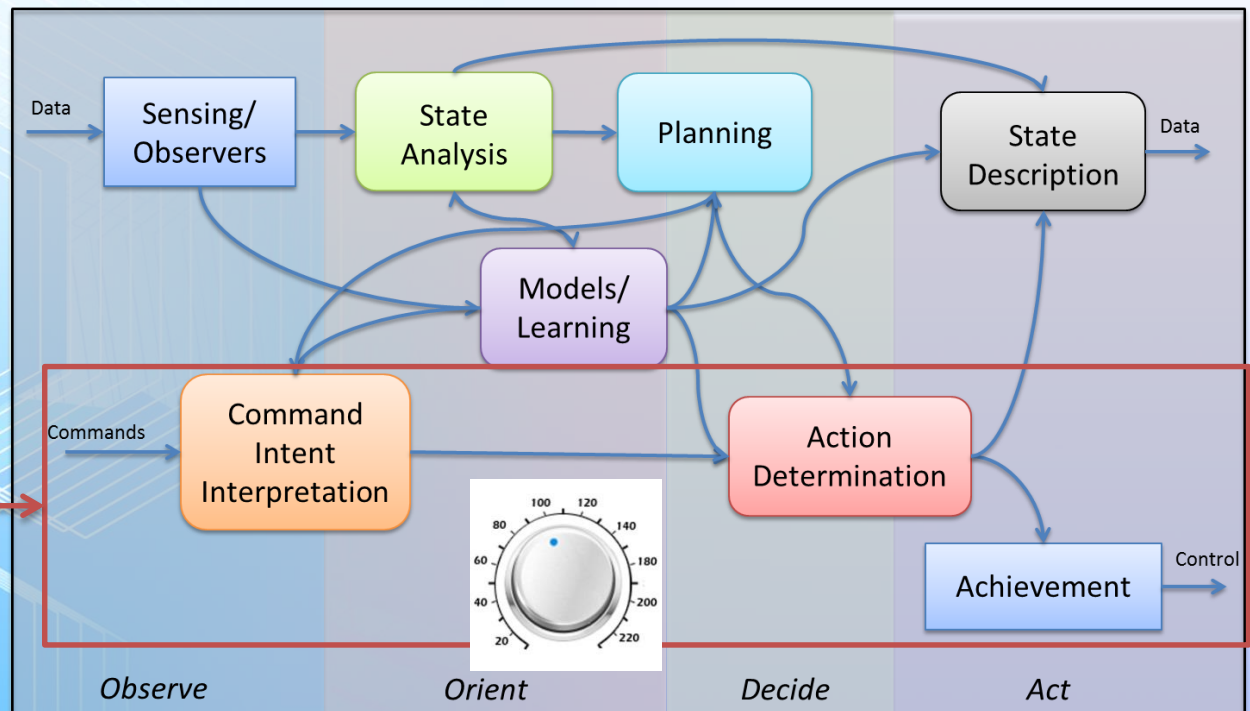


Modular Autonomous Systems Technology



- The Modular Autonomous Systems Technology (MAST) framework is an architecture that:
 - Can be used for all classes of autonomous systems
 - Standardizes information sharing and interfaces between technologies
 - Designed around formal verification and validation principles

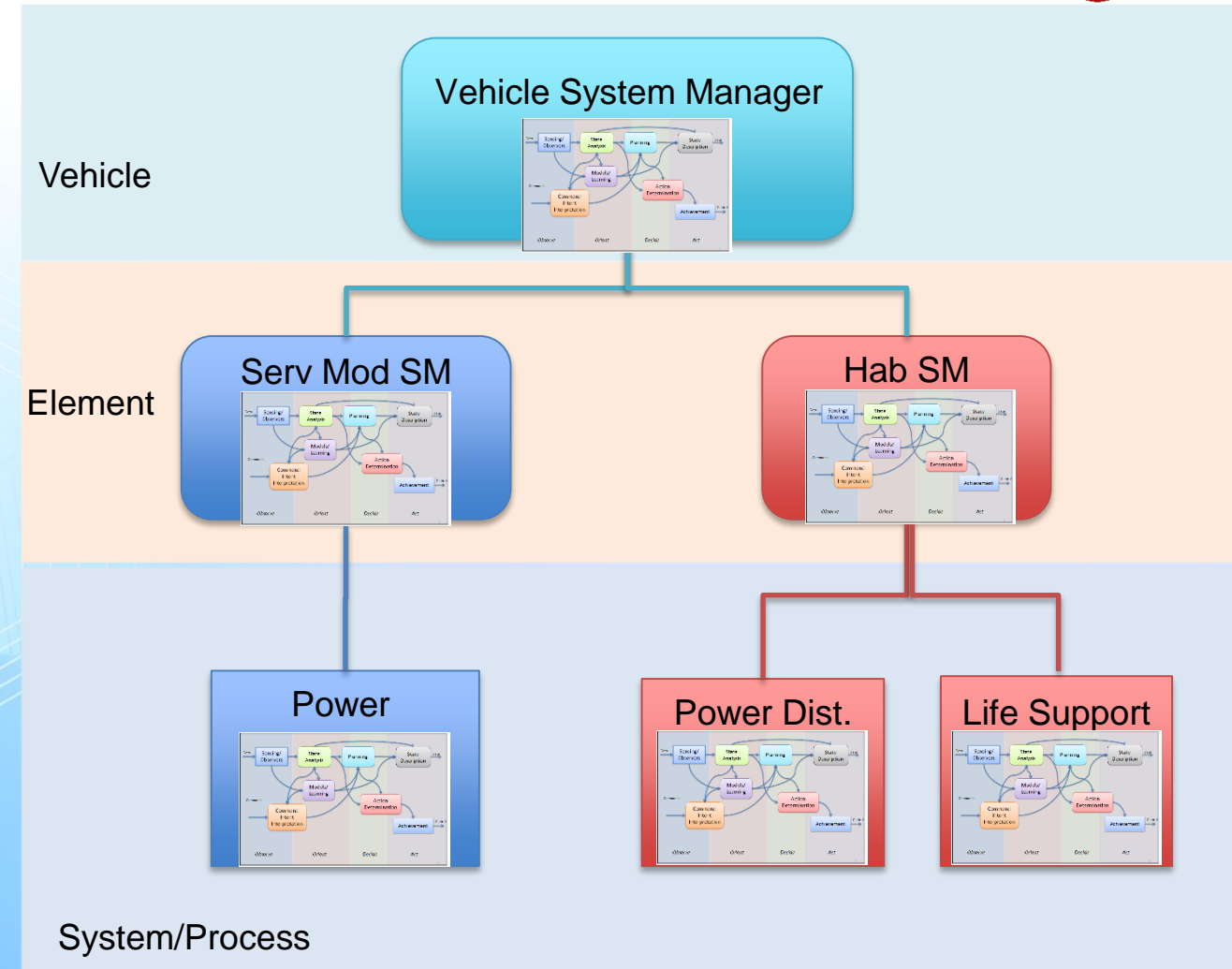
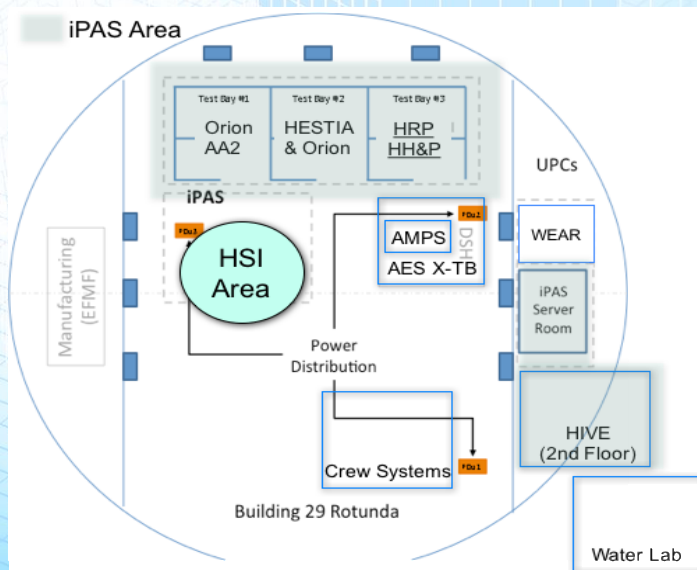
- Variable autonomy possible by figuring out how to throttle “actions”



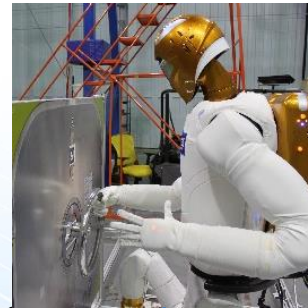
Autonomy Architecture Testing



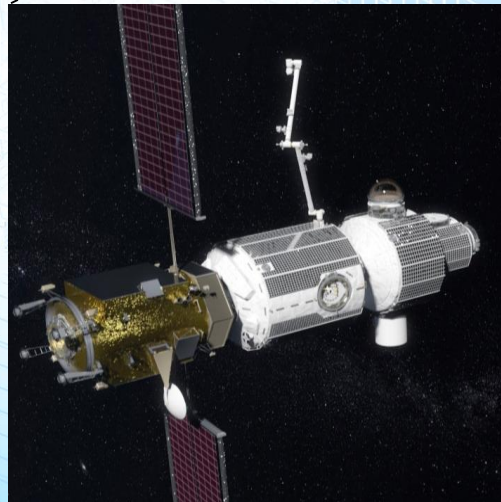
- **Developing a leak detection scenario**
 - Distributing autonomous functions
 - Using cognitive (learned) agents for detection
 - Testing architecture command/telemetry flow



Autonomy + Caretaker Roadmap



Manipulation	Generic Grasp Strategies	Expandable Object Ontologies	Robust Dexterity	
	Constrained Path Planning	Experience-based Planning	Task + Motion Planning	Localization
Mobility				
Sensing	Object Recognition	Object Localization	State Determination	Situational Awareness



Distributed Health Management	SHM under Uncertainty	Model Invalidation	Multi-agent State Determination	Prognostics
Task Planning	Planning under Uncertainty	Direct Translation of Activities to Tasks	Skill-based Multi-agent Task Planning	Distributed Planning & Execution
Data Management	Event Triggered Collection	Async. Distributed Sensing	Smart Downlink	Self-Directed Learning

Collaboration with Woodside Energy

Future Plans



The What

Advance the Robots

Influence the Environment

Smart Spacecraft

The How

1. Build new hardware
2. Build on AT for smart manipulation
3. Foray into task planning
4. **Demonstrate skills on orbit**

1. Define con ops for Gateway IVR
2. Outline a phased approach for increased robotic capability
3. Write IVR requirements

1. Incorporate planning and execution technologies
2. Understand data flow
3. Human interfaces and situational awareness

Applications

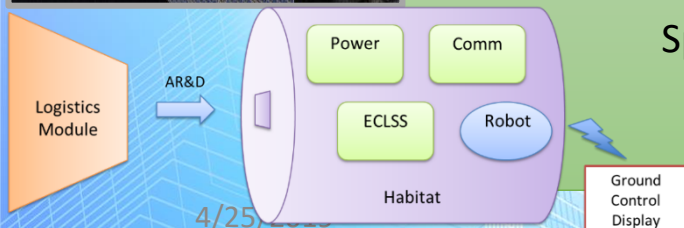
Logistics Management



Utilization



Maintenance



Conclusion

- Future exploration missions present unprecedented operational challenges
 - Robotics and autonomy will be key enablers of sustained human presence in deep space
- Interesting questions we hope to find answers to:
 - What roles can these technologies reliably play?
 - How does a system become trustworthy?
 - How do we design the system for optimal teaming with crew and ground operators?

