# CLARAty: Towards Standardized Abstractions and Interfaces for Robotics Systems

## Coupled Layer Architecture for Robotic Autonomy

Issa A.D. Nesnas

Jet Propulsion Laboratory,
California Institute of Technology

In Collaboration with

Ames Research Center
Carnegie Mellon University
University of Minnesota

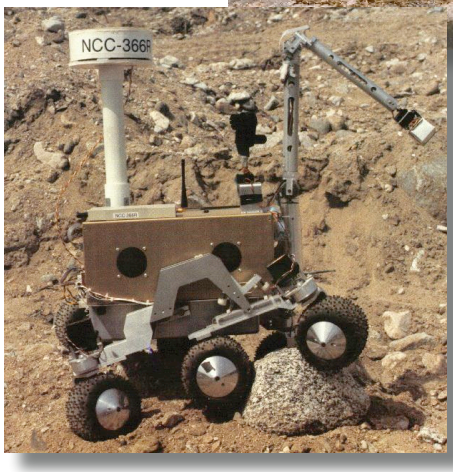
JTARS Meeting
May 11-12, 2005, Houston, Texas

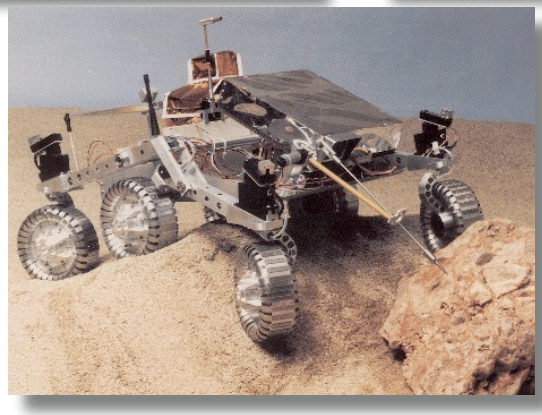
# Motivation

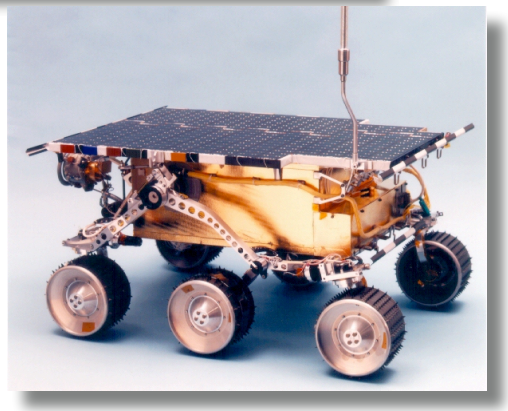
# NASA/JPL Develops Various Rovers

Large

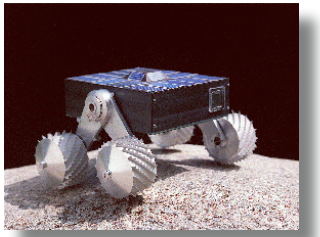
Medium



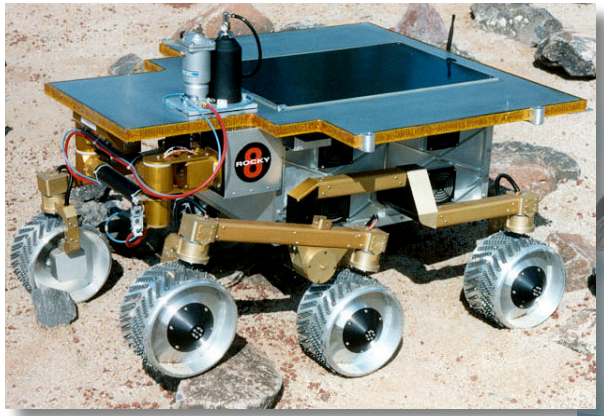


Small

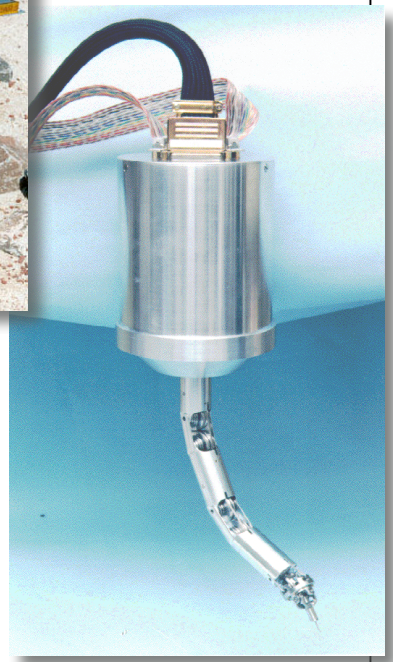


For research & flight

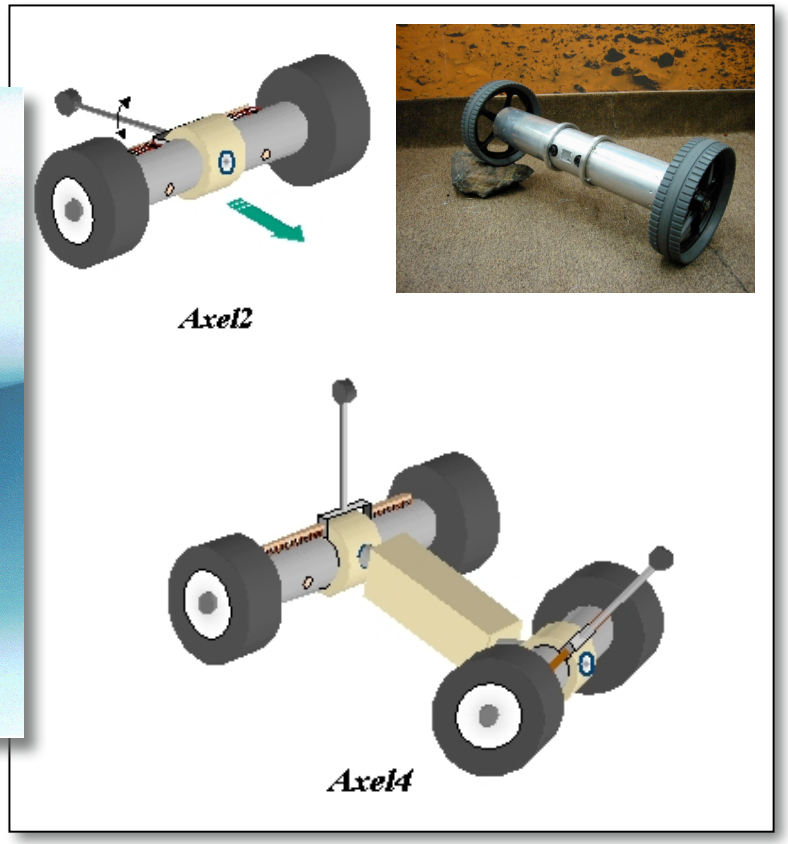
# Would like to support …



Custom Rovers



Manipulators




Reconfigurable Robots

COTS Systems

# Problem and Approach

- Problem:
  - Difficult to share software/algorithms across systems
  - Different hardware/software infrastructure
  - No standard protocols and APIs
  - No flexible code base of robotic capabilities

- Objectives
  - Unify robotic infrastructure and framework
  - Capture and integrate legacy algorithms
  - Simplify integration of new technology
  - Operate heterogeneous robots

# Challenges in Interoperability

- Mechanisms and Sensors
- Hardware Architecture

# Different Mobility Mechanisms

with different sensors

From wheeled Rocker-bogies with different steering

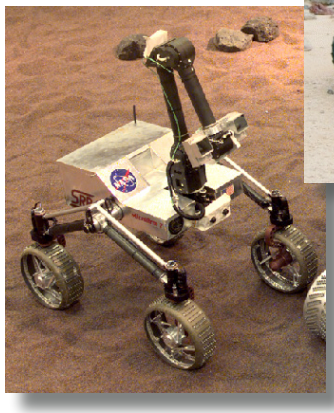
To wheels on articulated links

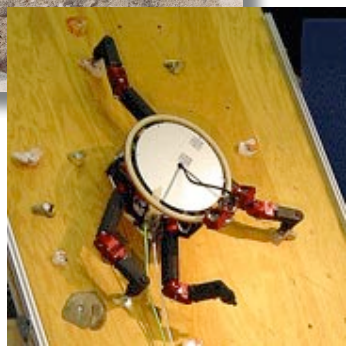
To inflatable wheels

From three wheelers

To four, six and even eight

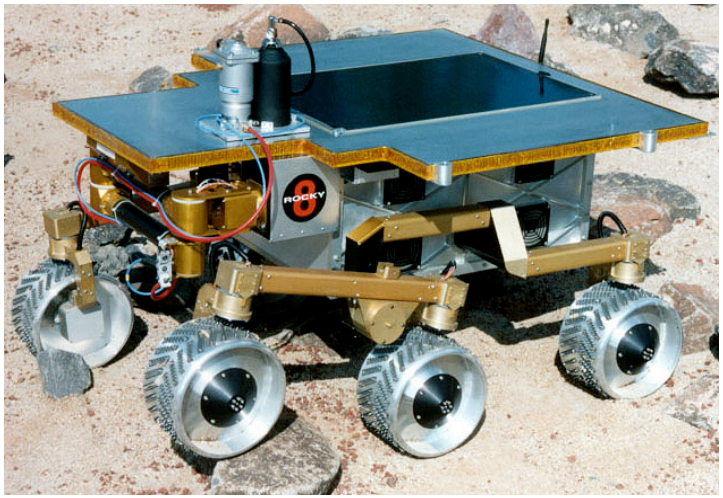
From wheeled to legged



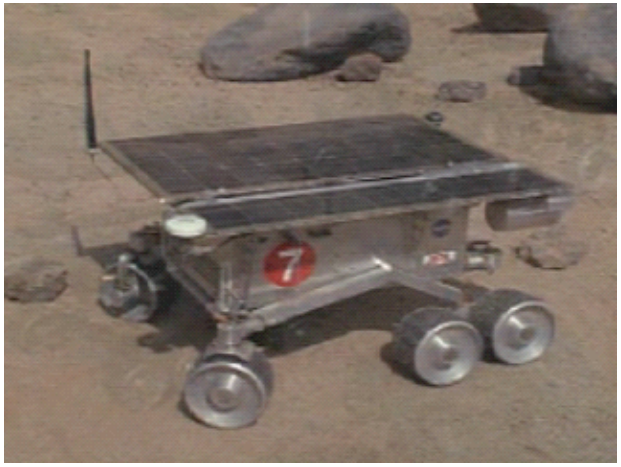


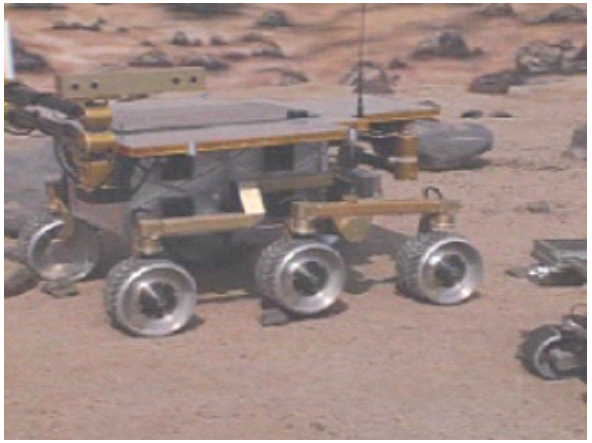
# For Example: Wheeled Locomotion

Rocky 7



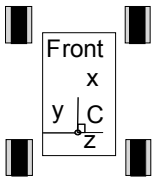
Rocky 8



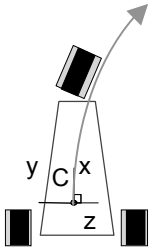


# Reusable Wheeled Locomotion Algorithms

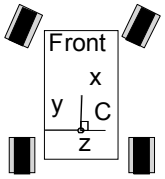
## General flat terrain algorithms and specialized full DOF algorithms



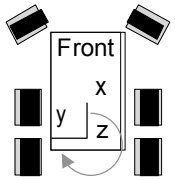

(a)
Skid Steering
(no steering wheels)



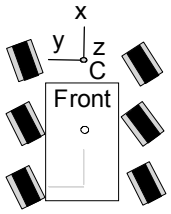

(b)
Tricycle
(one steering wheel)



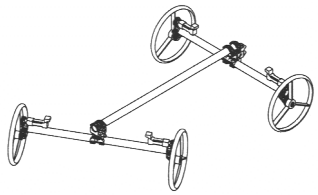

(c)
Two –wheel steering




(d)
Partially Steerable
(e.g. Sojourner,
Rocky 7)



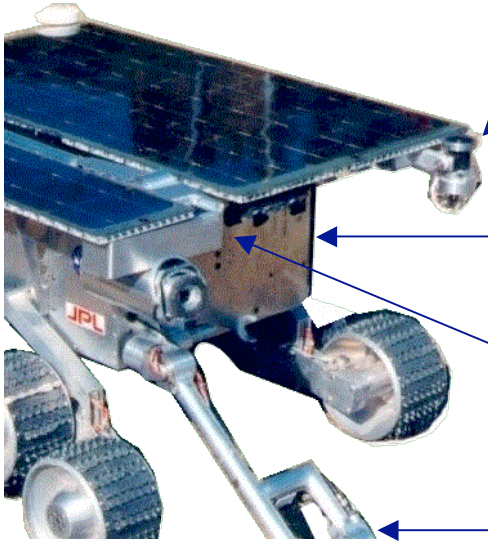

(e)
All wheel steering
(e.g. MER, Rocky8,
Fido, K9)

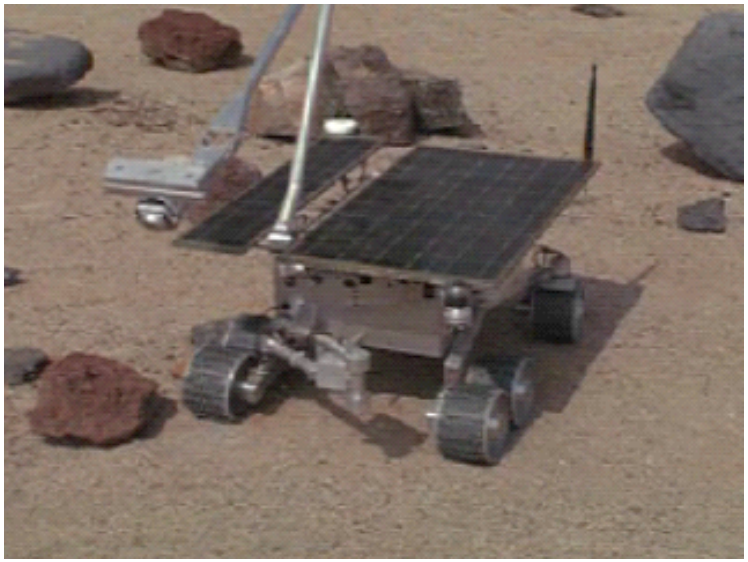


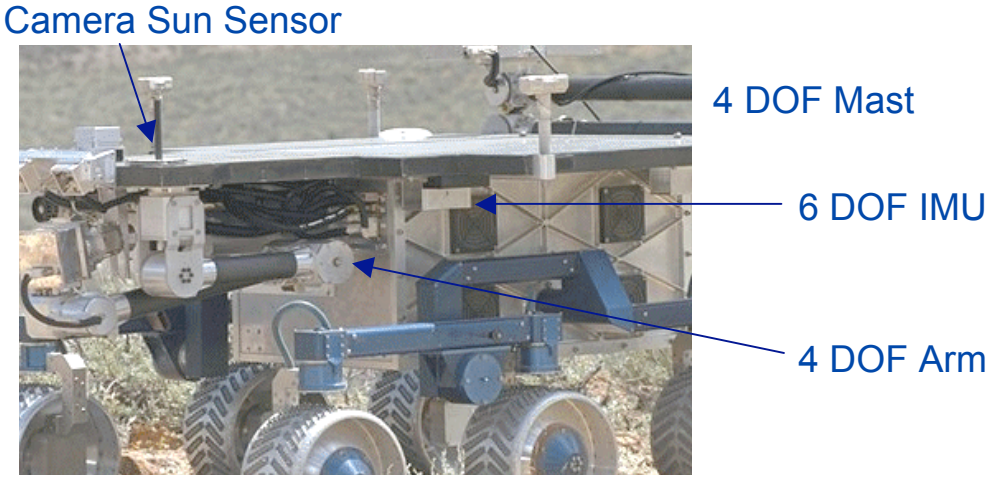
(f)
Steerable Axle
(e.g.Hyperion)

• • •

# Manipulators and Sensor Suites



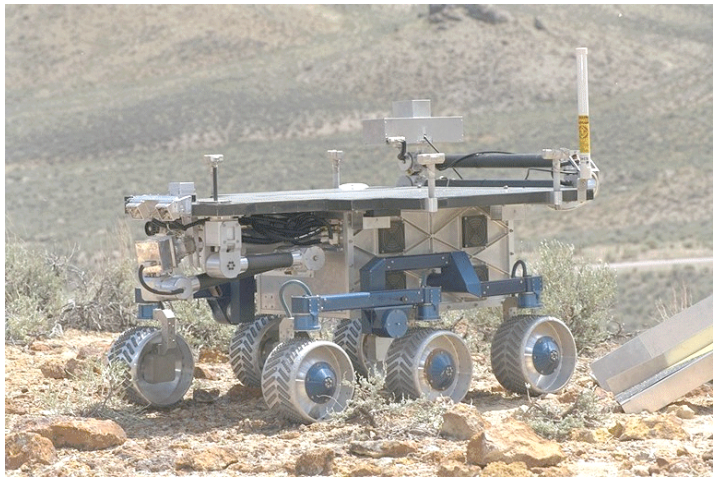



- Given different capabilities, how much reuse can be achieved?

# Challenges in Interoperability

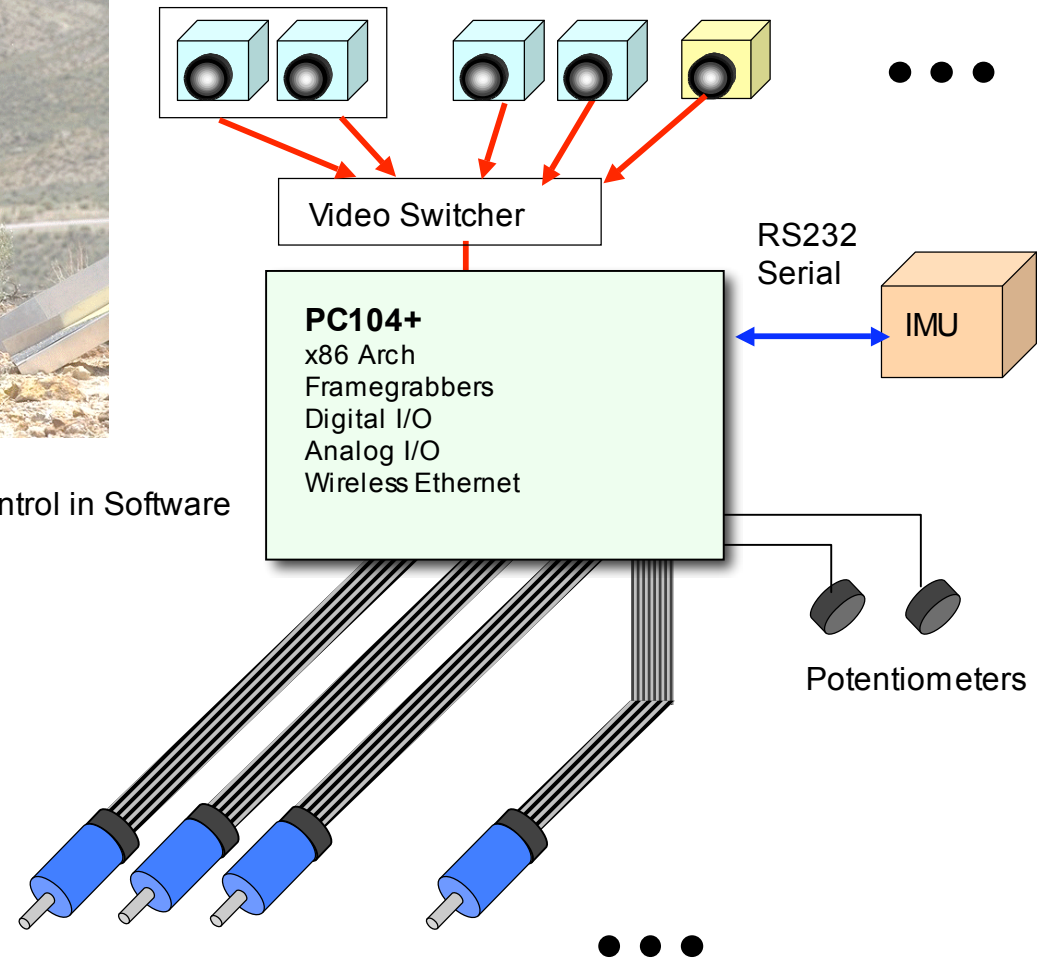
- Mechanisms and Sensors
- Hardware Architecture

# Centralized Hardware Architecture



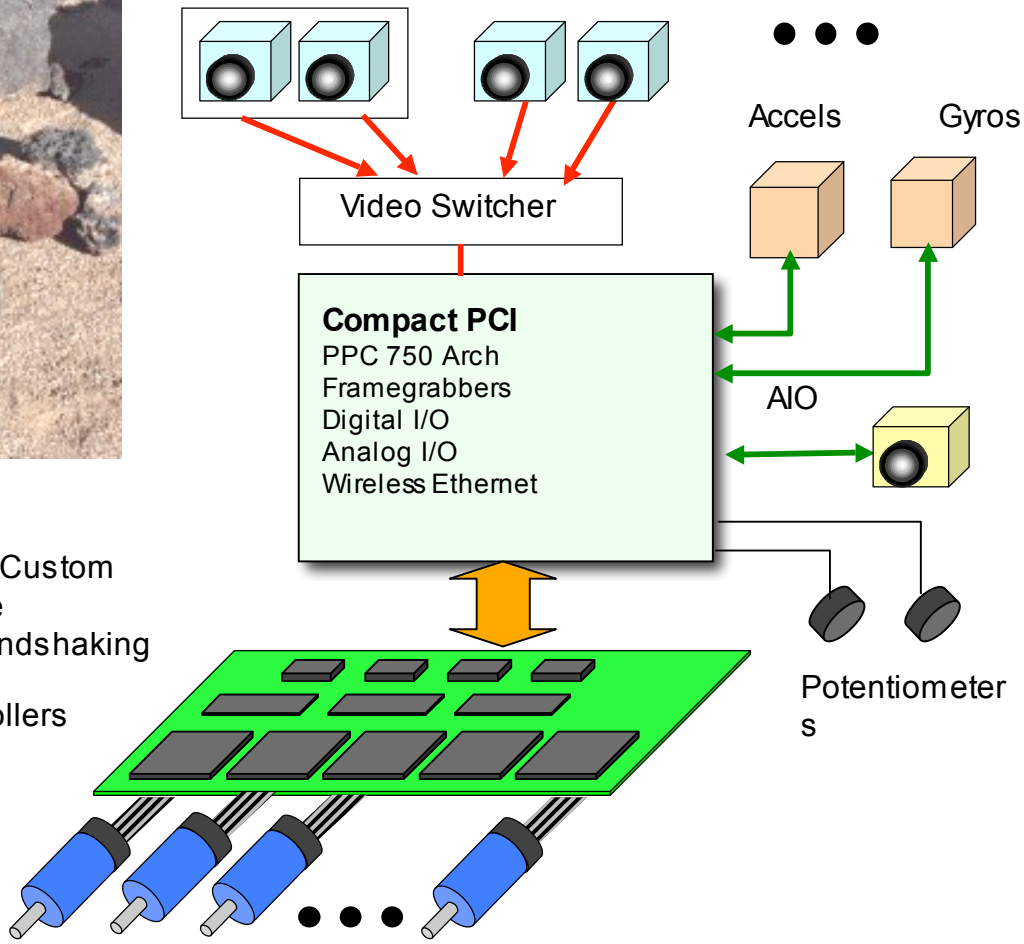
***FIDO***

PID Control in Software

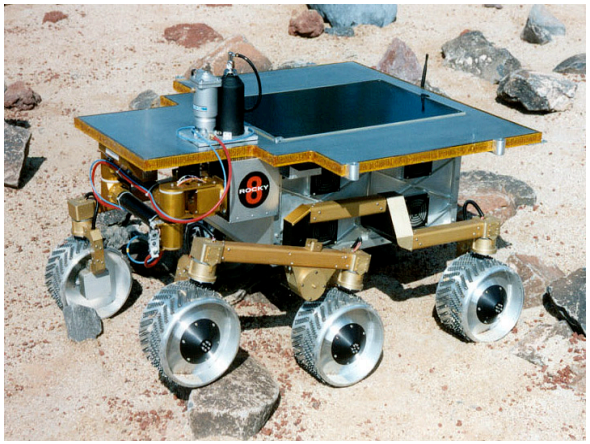


# Semi-centralized Hardware Architecture

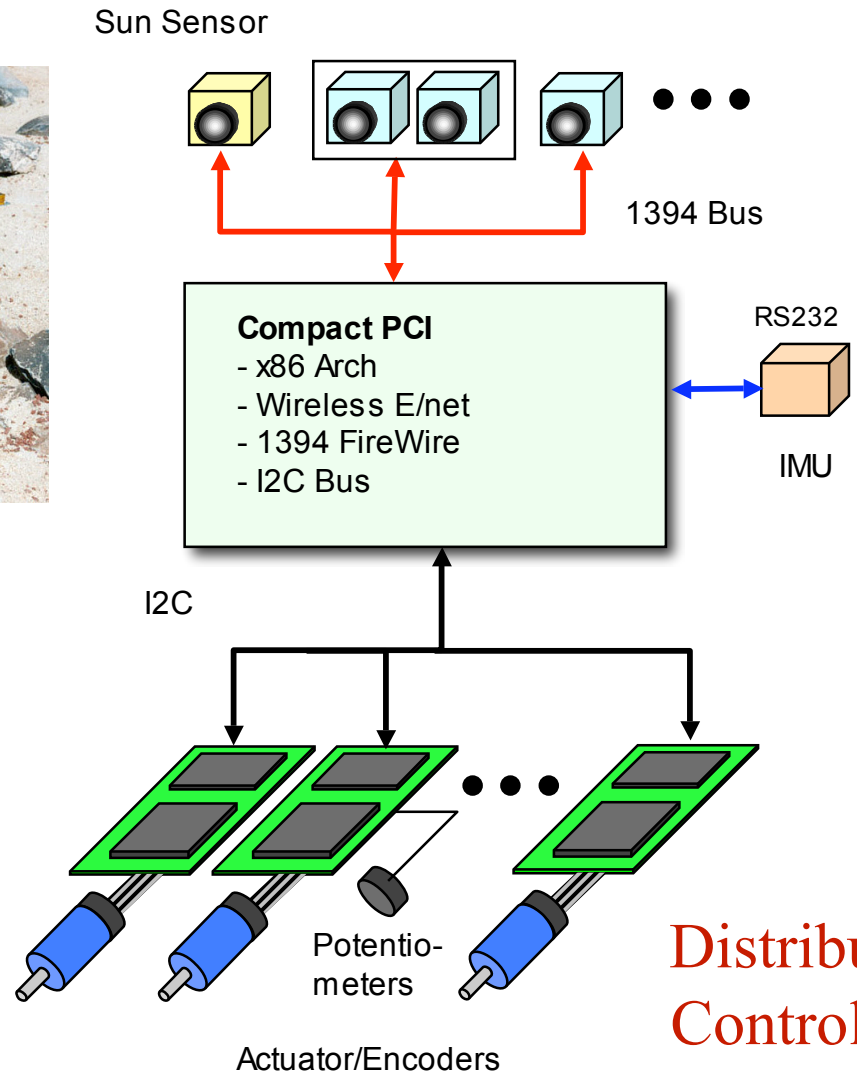
***Rocky 7***



# Semi-distributed Hardware Architecture



***Rocky 8***




***Rocky Widgets***
Single-axis controllers
Current sensing
Digital I/O
Analog I/O

Distributed Motion Control and Vision

# So what do we do?

# One Approach

Use the best attributes from each system and build a common platform

Unfortunately this is not always possible

OR

Develop a model to deal with the variability

# One Approach

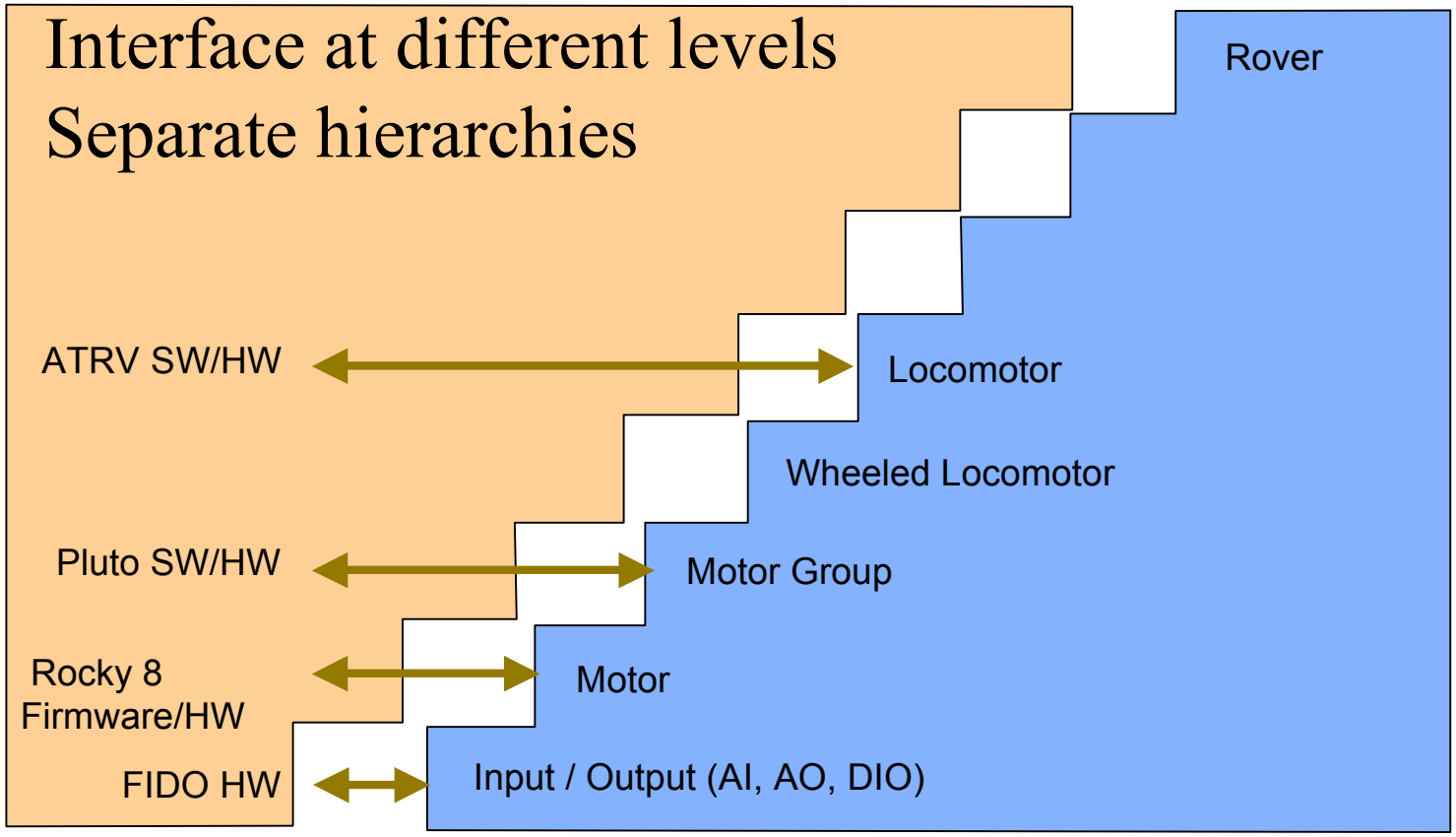
Use abstractions

Adapt as necessary

Interface at different levels

Separate hierarchies

Example of
Levels of Abstraction
for locomotors

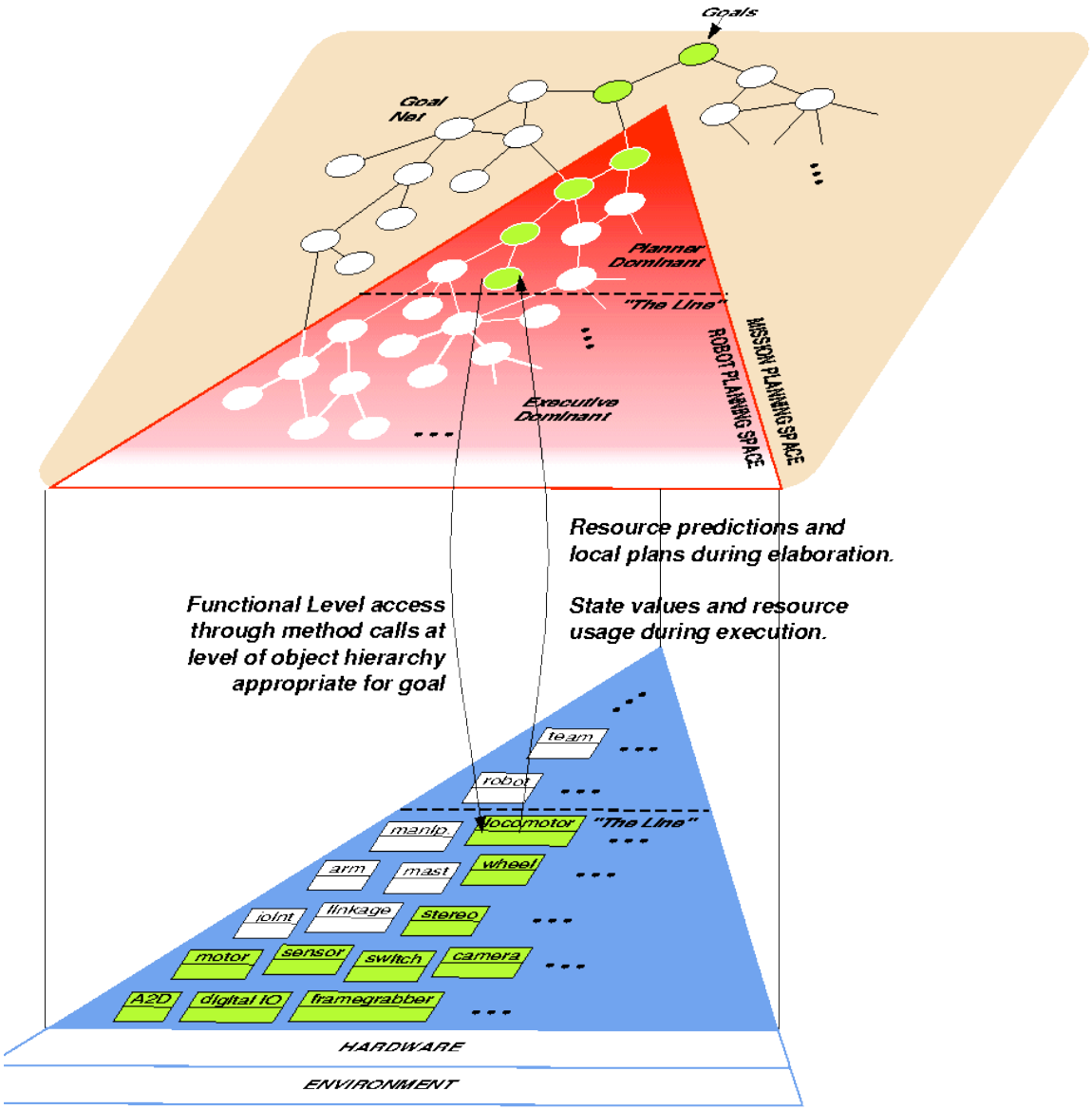


# One Approach

- Develop
  - Common data structures
  - Physical & Functional Abstractions
    - E.g. motor, camera, locomotor. Stereo processor, visual tracker
  - Unified models for the mechanism
- Putting it together
  - Start with top level goals
  - Elaborate to fine sub-goals
  - Choose the appropriate level to stop elaboration
  - Interface with abstractions
  - Abstractions translate goals to action
  - Specialize abstractions to talk to hardware
  - Hardware controls the systems and provide feedback

# A Two-Layered Architecture

CLARAty = Coupled Layer Architecture for Robotic Autonomy




***THE DECISION LAYER:***

Declarative model-based
Global planning

***INTERFACE:***

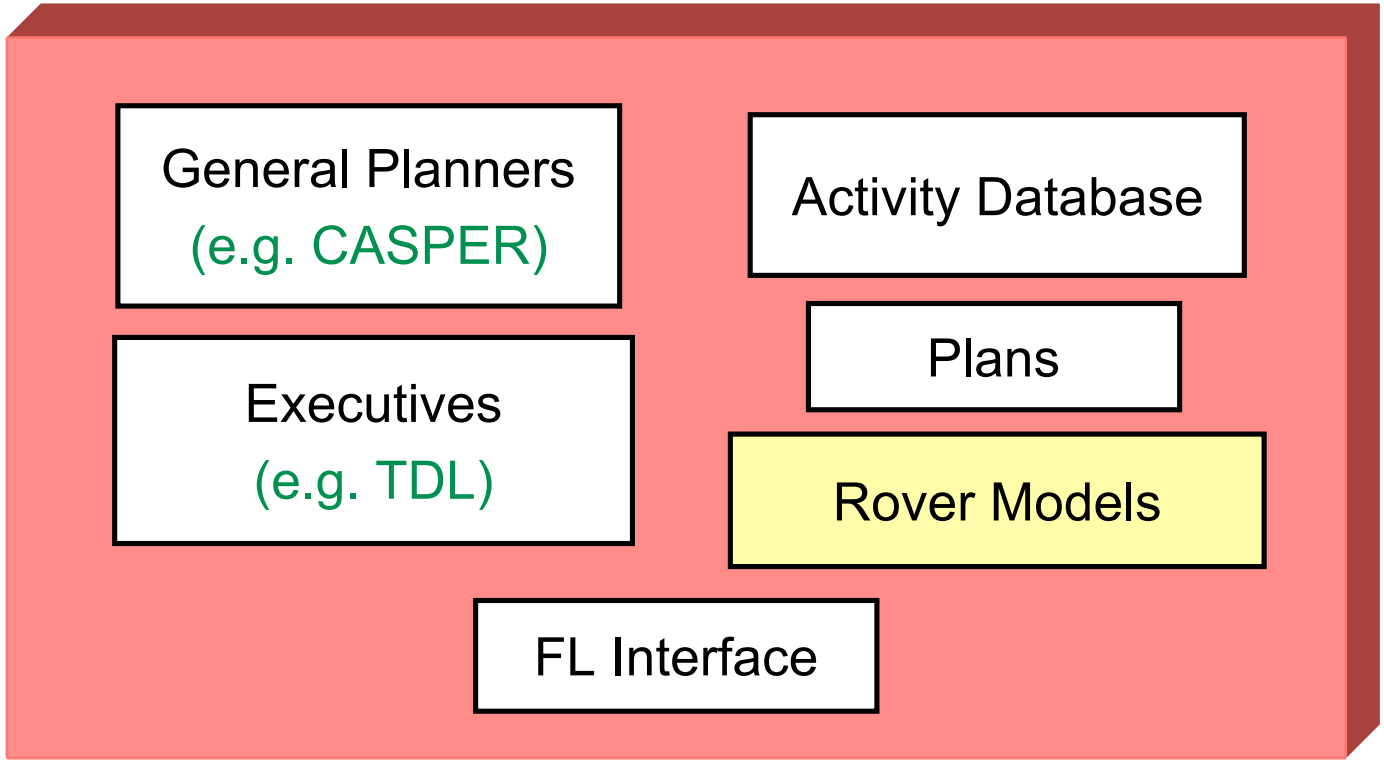
Access to various levels
Commanding and updates

***THE FUNCTIONAL LAYER:***

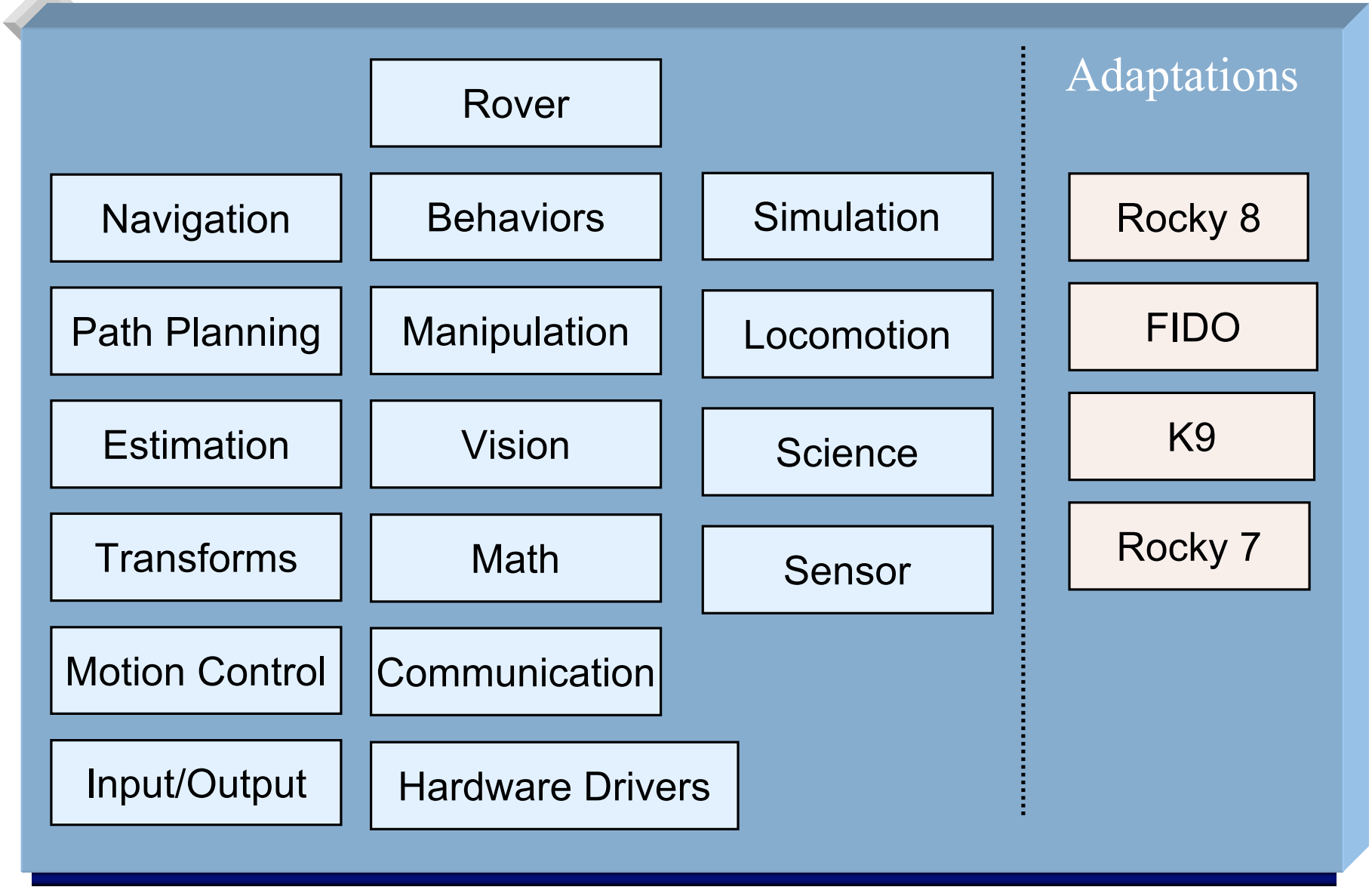
Object-oriented abstractions
Autonomous behavior
Basic system functionality

Adaptation to a system

# The Decision Layer



# The Functional Layer



# Standardizing Base Abstractions

## Abstractions

CoordMotionSystem

_position: Vector

Wheeled_Locomotor

Wheel

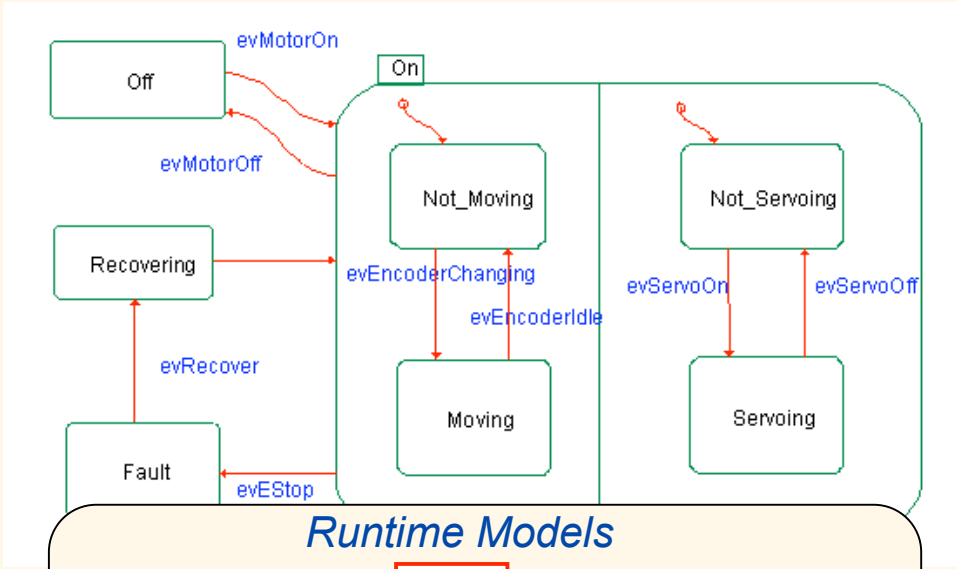
Steerable Wheel

## APIs and Behaviors

Wheel Locomotor

-Vehicle_Interface & _interface
-DPoint _goal
-double _goal_heading

## State Machines




## Runtime Models

Motor

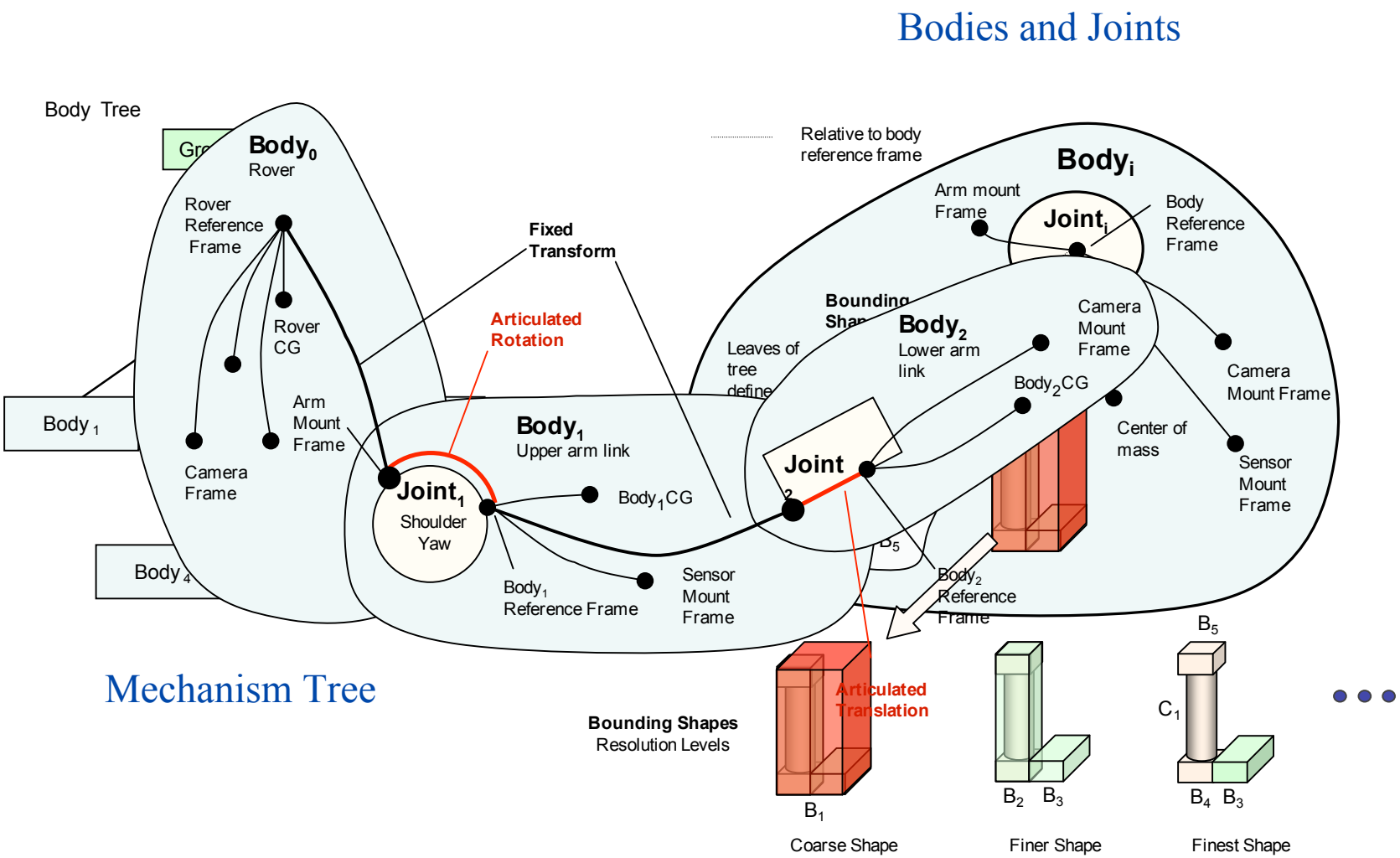
Thread 1 (controls)

```
motor.change_position(2PI)
do other things
```

Thread 2 (monitors)

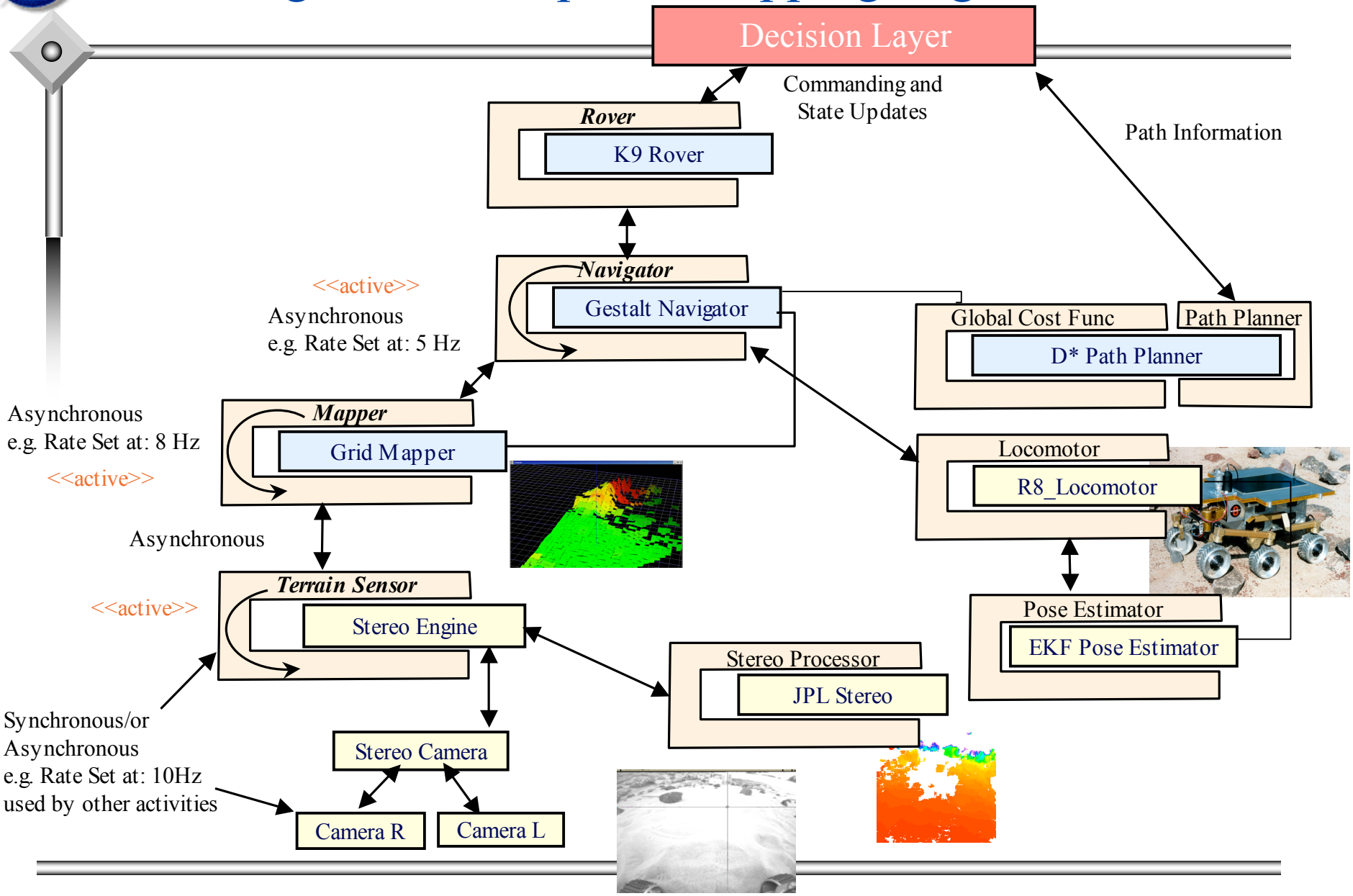
```
based on watchdog
position = motor.get_position()
if (position > x) motor.stop()
```

# Unify Mechanism Model

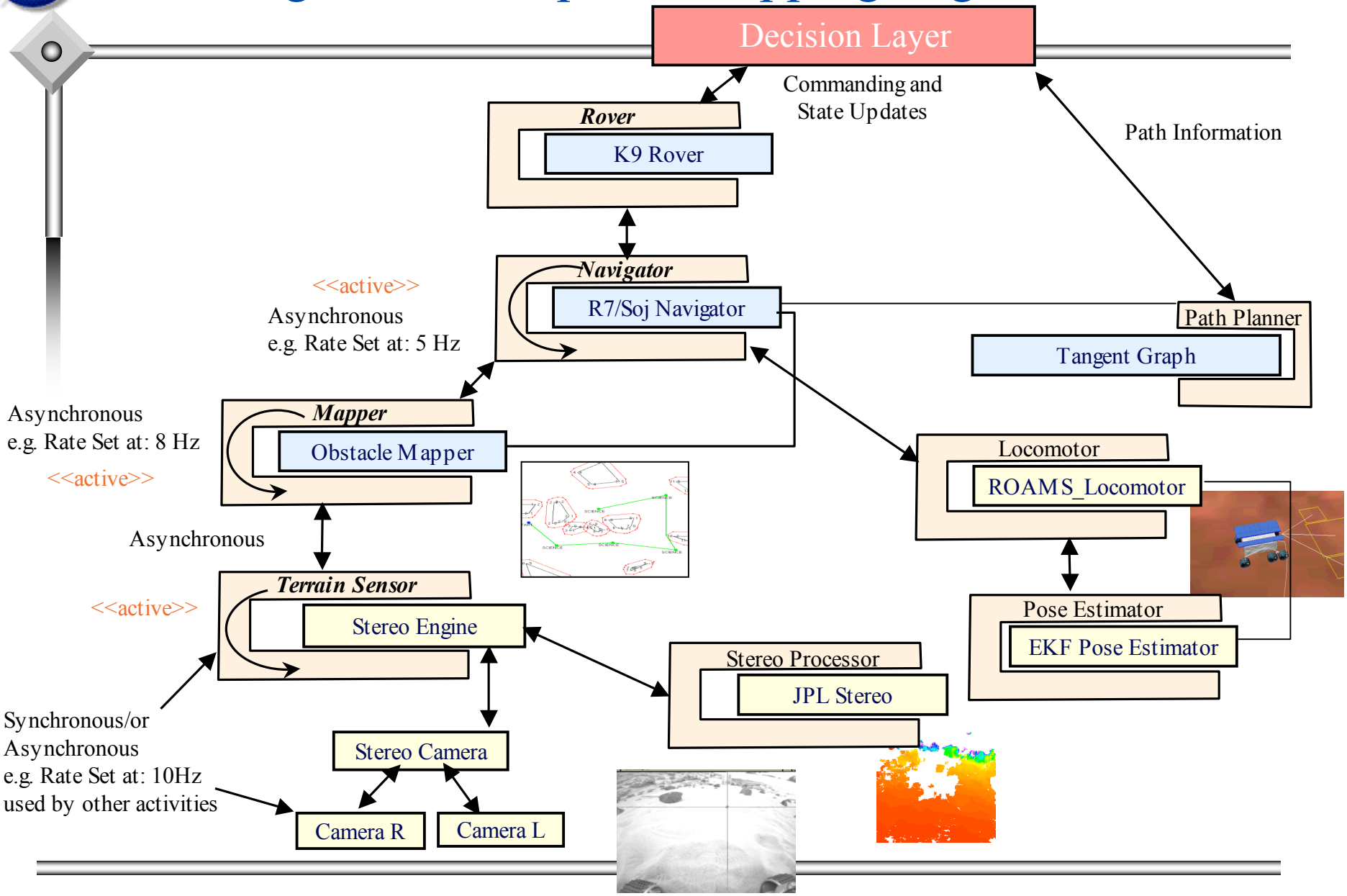


# Putting it All Together

# Navigation Example - Swapping Algorithms



# Navigation Example - Swapping Algorithms



# Navigation with Path Planning on Two Rovers

**Complex Algorithms on different Platforms**

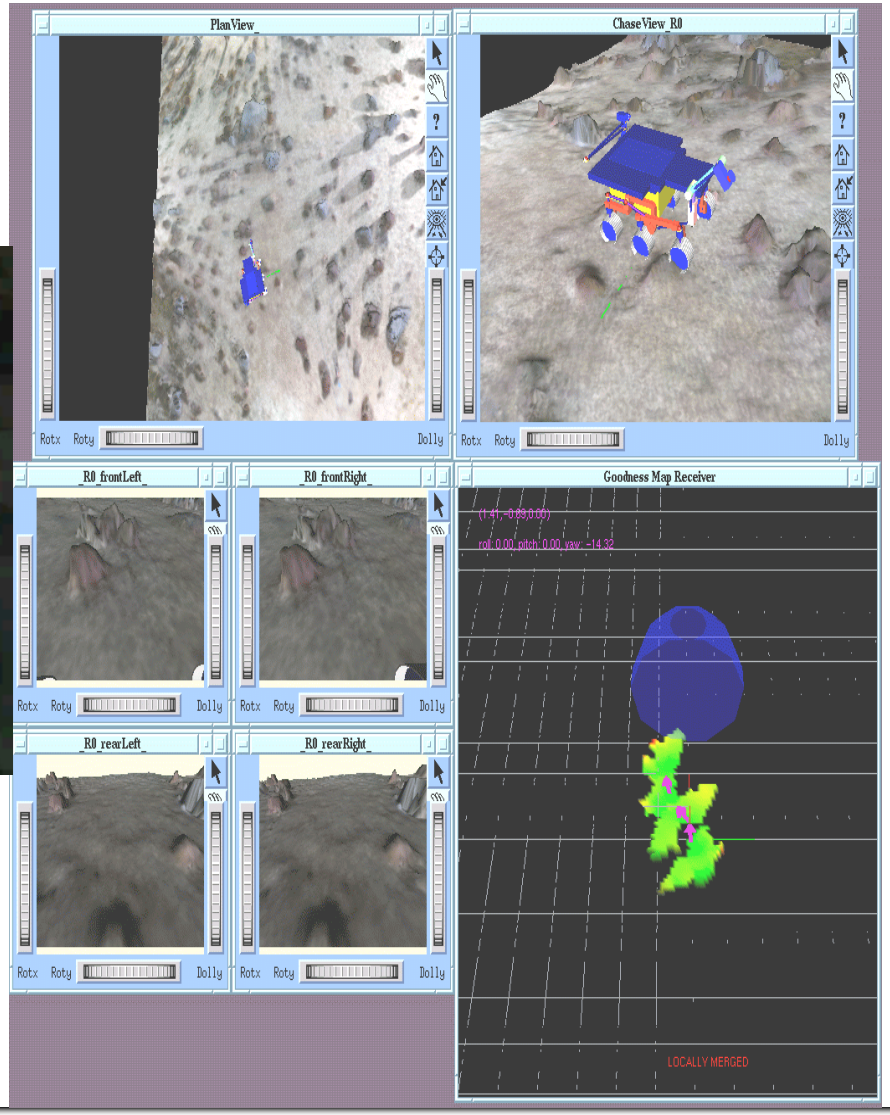
- I/O, motion control
- Trajectory Generation
- Rough Terrain Locomotion
- Odometry Pose Estimation
- Stereo Processing
- Visual Odometry
- Navigation (Morphin)
  - Obstacle avoidance
  - Path Planning

# Designated Target Tracking for Single-Cycle Instrument Placement

**Integration of Complex Algorithms**

- I/O, motion control
- Trajectory Generation
- Rough Terrain Locomotion
- Odometry Pose Estimation
- Stereo Processing
- Visual Odometry
- *Obstacle avoidance*
- Mast Control
- Visual Tracking

# And with a Simulated Rover



# Acknowledgements

CLARAty Team (multi-center)

**Jet Propulsion Laboratory**

- ROAMS/Darts Team
- CLEaR Team
- Instrument Simulation Team
- Machine Vision Group
- Robotic Systems Group

**Ames Research Center**

- K9 Team

**Carnegie Mellon University**

**University of Minnesota**

*Thank you for your Attention*

# *Backup Slides*