

ASTRA 2015  
Robot Control Operating System (RCOS) Forum

***System and Programmatic Issues  
In RCOS Design***

Richard Volpe

*Manager, Mobility & Robotic Systems Section  
Jet Propulsion Laboratory  
California Institute of Technology*

*12 May 2015*

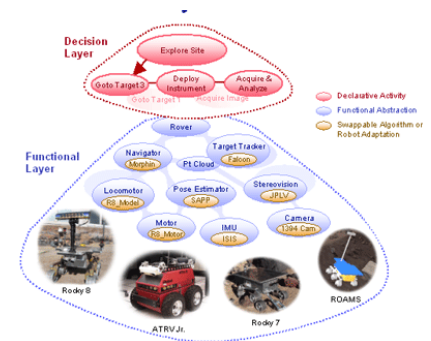


*Copyright 2015 California Institute of Technology.  
Government sponsorship acknowledged.*

It is illustrative to review the system and programmatic objectives that have driven example RCOS system development over the years.

Relevant examples to be discussed:

1. ~1990: Chimera II at CMU with DD Arm II
2. ~1997: RTI ControlShell with VxWorks for JPL for Rocky 7
3. ~2005: CLARAty with VxWorks for NASA's Mars Tech Program
4. Today: RSAP with Linux on JPL's Robosimian



This talk will not discuss the details of the implementation, but rather the forces that shaped the chosen solutions.

## 25 years ago... a solution for manipulator control research at Carnegie Mellon.

[Stewart, Khosla, Volpe, et al.]

**Objective:** Develop a modular and reconfigurable control system infrastructure to support manipulation controls research, primarily on one robot, CMU Direct Drive Arm II.

### Requirements and Desires:

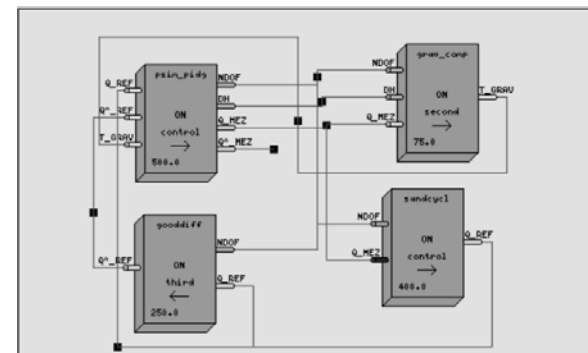
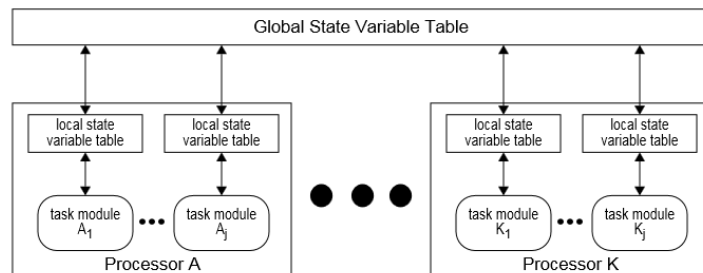
- Provide a **Real Time Operating System (RTOS)**, since there was none to leverage
- Provide **shared memory management** for multi process/processor communication.
- Provide **device abstraction** for ease of controls sensor integration
- Make **modular and reconfigurable** at run-time for productivity of research, and control mode switching.

### Chimera Features:

- **State variable** management and data transport infrastructure
- Modular and reconfigurable **port-based objects**.
- Command-line or graphical **configuration control**.
- Selectable **task scheduling** infrastructure
- Support for **multi-process, multi-processor**, and co-processors.

### Chimera Issues:

- Limited product support and porting to new hardware
- Limited user community



**18 years ago... Leveraging commercial solutions for Research Rover System Development** [Volpe, Hayati, Balaram, et al.]



**Objective:** Use a modular and reconfigurable control system infrastructure to support Mars Rover Technology System development and demonstration, primarily on one robot, Rocky 7.

**Requirements and Desirements:**

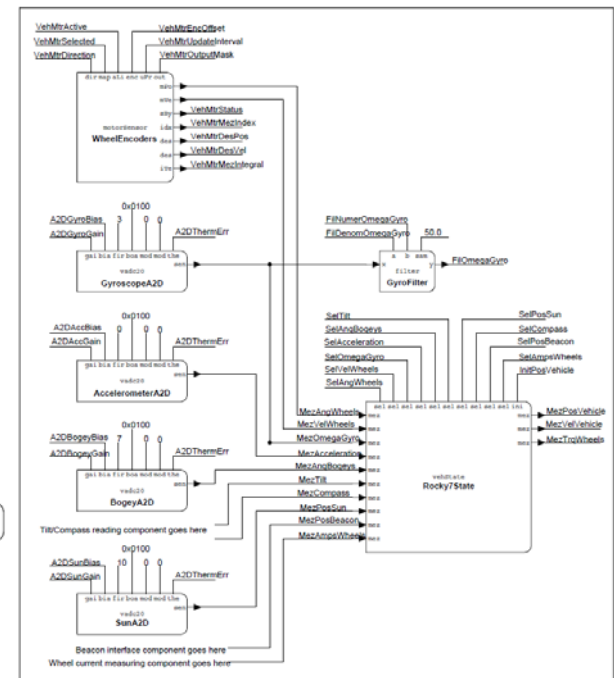
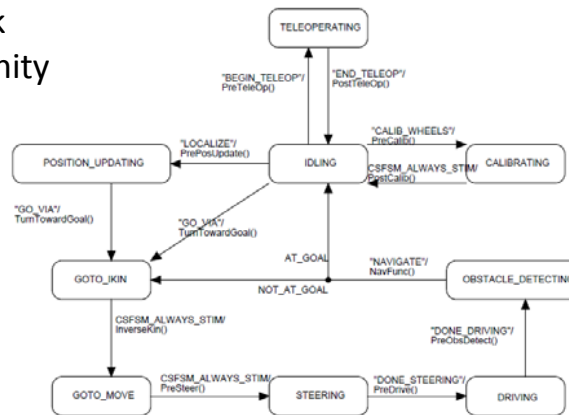
- Leverage commercial software, with **promise of greater reliability and stability** than home grown solutions
- Allow research team to concentrate on system development and controls issues, and **not be distracted by infrastructure.**
- Augment controls with **state machine infrastructure** to allow execution of mission scenarios.

**ControlShell/VxWorks Features:**

- Modular, run-time reconfigurable software
- Graphical interface for system configuration
- Hierarchical State Machines
- Flight-certified RTOS

**ControlShell/VxWorks Issues:**

- VxWorks is expensive for a research task
- ControlShell had a limited user community



## 10 years ago... Unite the NASA robotics community for Mars Technology Development [Nesnas, Volpe, et al.]

**Objective:** Develop Mars rover technology for robotics and AI using a common software platform across heterogeneous rovers and multiple research centers and universities.

### Requirements and Desires:

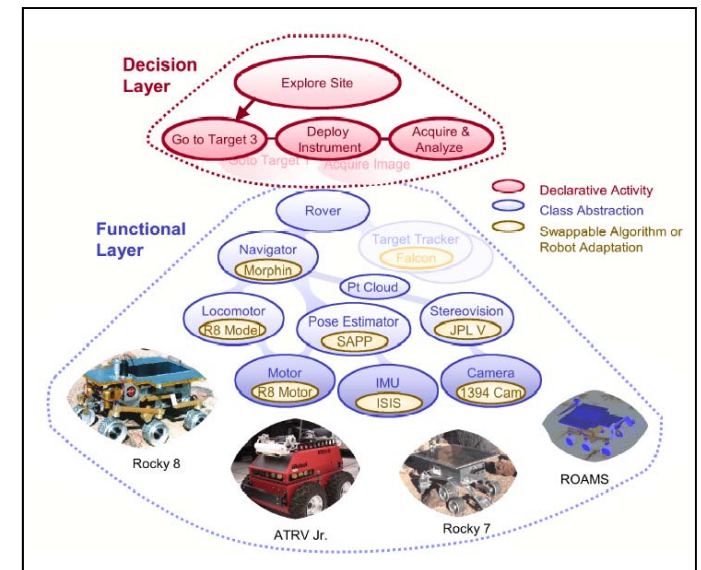
- Support **heterogeneous systems** from hardware to algorithms.
- Support **interoperability of different algorithms** for a given capability.
- Facilitate the integration of **robotic control with AI systems**, particularly planning and scheduling.
- Leverage **object-oriented design** principles.
- Strong support from a major sponsor, the **Mars Technology Program**

### Coupled-Layer Architecture for Robotic Autonomy (CLARAty) Features:

- **Variable levels of abstraction** for hardware and algorithms.
- Supported a **large development and user team** across NASA and Academia (primary partners were JPL, NASA Ames, and CMU).
- Provided **common language and code base** for robotics and AI researchers.
- Enabled **integration and maturation** of algorithms on heterogeneous platforms with validation resulting in **flight infusion**.
- Portions of the code have been publically released.

### CLARAty Issues:

- Primary sponsor has removed development support due to budget cuts
- User community has dwindled.
- Costly to maintain across heterogeneous platforms.

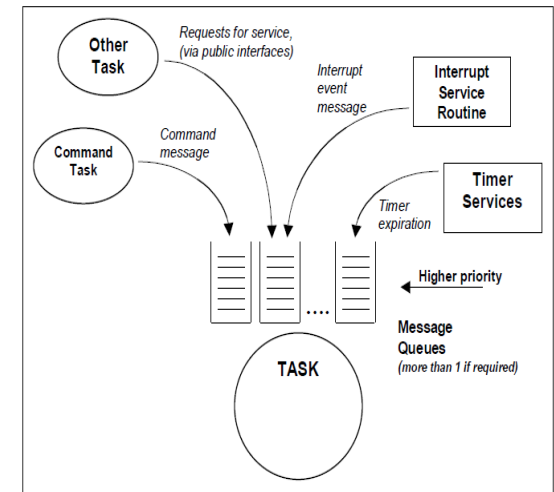
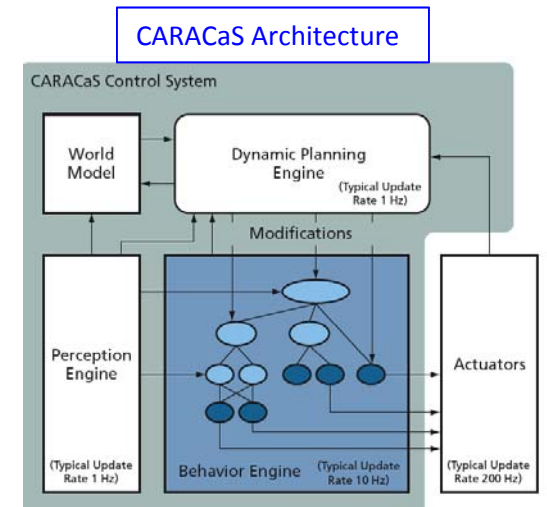


## Today... Different Solutions for Different Customers

**Objective:** Leverage different solutions that are appropriate for different customers, due to constraints such as heritage, cost, ITAR, program requirements, etc.

### Example Solutions:

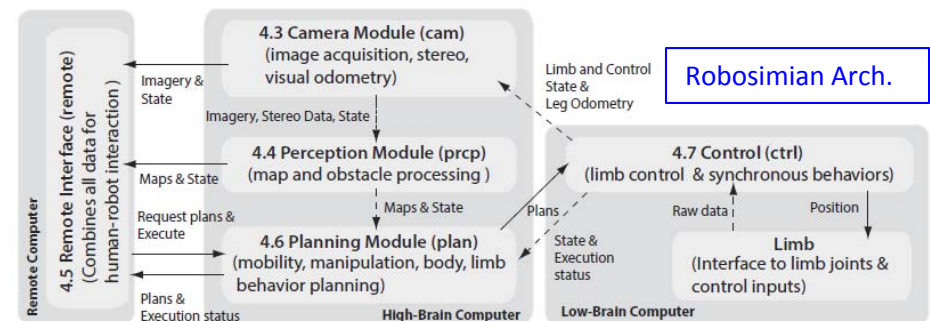
- Various DoD programs: JAUS, ROS, etc.
- US Navy Programs supported by JPL robotics:
  - VxWorks and QNX
  - R4SA (Robust Reconfigurable Real Time Robotics Software Architecture) *lightweight functional level (FIDO Project derivative, ~2000)*
  - CARACaS (Control Architecture for Robotic Agent Command and Sensing) *perception system, behavior engine, and dynamic planner*
- DARPA Robotics Challenge (Robosimian)
  - RSAP (Robotics Software Architecture Platform) *similar to flight robotics software architecture, but ITAR-free.*
- Robotics Flight Software (MER - Mars Exploration Rovers)
  - Multi-tasking & message passing. See: *Reeves, G., & Snyder, J. 2005. An overview of the Mars exploration rovers flight software. IEEE Conference on Systems, Man, and Cybernetics, Big Island, HI.*



Navy boats

Robosimian

MER



Robosimian Arch.

**Conclusions:**

1. Customer focus will drive the product
2. Heritage software is hard to ignore. (For the developer and the customer.)
3. A diversity of solutions exist and the set is not converging (yet) the way other software solutions have converged (e.g. desktop environments).