Mars Rover Research and Software Infrastructure Development With CLARAty

Richard Volpe
JPL Space Exploration Technology Program Office
NASA Mars Technology Program
2009 Mars Science Laboratory Project

Issa Nesnas
Mobility Concepts Technology Section

Jet Propulsion Laboratory
California Institute of Technology
Outline

- Mars Rover Technology History Lessons
- CLARAty Overview
- Multi-Platform Support
- Multi-User Integration
- New Technology Validation and Infusion
- Summary
JPL Rovers, Technology History

1990, Robby: stereo vision, 6 DOF manipulation, path planning, conditional sequence execution, Sun workstation and 68020 processors. (3m long, 1000kg mass) 100m traverse in arroyo.

1994, Rocky 4: bump/proximity sensing, 1 DOF rock chipper, reactive navigation algorithm, integrated spectrometer, 6811 processor. (60cm, 10kg). Tens of meters traverse in arroyo with soil return to lander with active beacon

1997, Rocky 7: stereo vision, 4 DOF arm, 3 DOF mast, stereo vision for onboard hazard detection and path planning, sun sensing and gyro for heading determination, rock constellation based localization. 68060 processor, (60cm, 15kg). 1km integrated traverse in Mojave.

2001, FIDO: stereo vision, 4 DOF arm, 4 DOF mast, 3 DOF drill positioning, stereo vision for onboard terrain traversability measurement, sun sensing and IMU for position estimation, PII processor, (1m, 50kg). 20 sol desert operations from JPL including traverse and instrument positioning.
Taxonomy of Mars Rover Navigation Development


Local 'Bug' Navigation

Rocky 4 ————- Sojourner

Rover Bug ————- Fuzzy Nav

Local Grid-Map Navigation

Ranger ————- Morphin

Morphin’ ————- GESTALT

Global Path Planning

D* ————- TEMPEST

DriveMaps ————- ROAMAN

** Note that integration typically requires refactoring of navigation systems into parts (e.g. maps, navigation, vision, etc.)
Improvements in Mars Rover Position Determination

Legacy Techniques

1. Odom. + gyrocompass
2. Odom. + compass
3. Odom. + angular rate sensor
4. Odom. + IMU + sun sensor
5. Odom. + IMU + EKF (+ ss)

Acc. | Robby | Rocky 4 | Sojourner | Rocky 7 | FIDO

New MTP Products

6. Visual Odometry
7. Full Kinematics
8. Sinkage Estimation
9. Slippage Estimation
10. Full Estimator
So, what are the impediments to progress?

Duplicative efforts prevent attainment of critical mass:

- **Parallel Duplication**: Many mobile robot projects within NASA funded institutions are building systems of similar functionality without sharing the burden of software infrastructure development.

- **Serial Duplication**: New starts of projects often wipe the slate clean to eliminate old system problems and lack of familiarity or trust with previous product. Typically, software with legacy is due solely to a single individual or team, not the community.

Need to Follow software community lead:

- **Open source movement**: The value of shared software is illustrated by Linux, GNU, Intel Computer Vision Library, etc.

- **Object oriented design**: It dominates software development, especially in industry, but is under-utilized in robotics.

Leveraging complimentary efforts:

- **Software sharing**: Across related tasks within NASA (and DoD) is often arduous and rare.
Unified Research Software Motivation

1. Legacy Capture
2. Tighter Coupling of Robotics & AI
3. Multi Platform Support
4. Multi User Integration
5. Complementary Algorithm Leveraging
6. Competitive Algorithm Comparison
7. Technology Validation
8. Mission Infusion
Enabling Architecture

CLARAty = Coupled Layer Architecture for Robotic Autonomy
A View of Architecture Hierarchy

Typical 3 Level Architecture

- Functional Level is often flat — typically a thin layer over the hardware
- Planner has no access to Functional Layer.
- Abstraction and granularity is mixed with intelligence.

CLARAty 2 Layer Architecture

- Functional Layer contains object-oriented abstraction of hardware at all levels of system granularity.
- Planner and Exec are similar, dominating at different levels of granularity, sharing a common database.
- Planner does not have direct access to the Functional Layer for execution, but executive may be minimized.
CLARAty Two-Layered Architecture

CLARAty = Coupled Layer Architecture for Robotic Autonomy

THE DECISION LAYER:
Declarative model-based
Mission and system constraints
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
Granularity Matching between Layers of Decision, Function, Simulation, and Operations

Key points:
- Granularity penetration at each level may vary.
- Simulation may exist at all levels of granularity.
- Operations access to Functional Layer through DL, even if at a single point.
- Decision Layer access to simulation passes through Functional Layer, even at a single point.
- DL / FL interface independent of simulation or real system.
CLARAty
Functional Layer:
Object-Oriented Hierarchy

- Hardware
- Environment

Robot

Motor

Coordinated System

Appendage

Locomotor

My Rover

My Rover's Locomotor

My Rover's Arm
FL Architectural Components

- Generic Physical Components (GPC)
  - Locomotor, Arm, Mast, Spectrometer, ...
- Specialized Physical Components (SPC)

- Generic Functional Components (GFC)
  - ObjectFinder, LayerDetector, VisualNavigator, StereoProcessor, ...
- Specialized Functional Components (SFC)

- Data Structure Components (DSC)
  - Array, Vector, Matrix, Map, Container, LinkedList, Bit
  - Image, Message, Resource
Multi Platform Support
Currently Supported Platforms

Rocky 8
VxWorks  
x86
JPL

K9
Linux  
x86
Ames

Rocky 7
VxWorks  
ppc
JPL

FIDO
VxWorks  
x86
JPL

ATRV
Linux  
x86
CMU

ROAMS
Solaris  
Linux
JPL
Distributed Hardware Architecture

**Rocky 8**

*Widgets*
- Single Axis Controllers
- Current Sensing
- Digital I/O
- Analog I/O

Actuators/Encoders

Potentiometers

Sun Sensor

1394 Bus

12C Serial Bus

Compact PCI
- x86 Arch
- Wireless ethernet
- 1394 FireWire
- I2C Bus

IMU

RS232 Serial
Centralized Hardware Mapped Architecture

Fido

PID Control in Software

Actuator/Encoders

PC104
x86 Arch
Framegrabbers
Digital I/O
Analog I/O
Wireless ethernet

Video Switcher

RS232 Serial
IMU

Potentiometers
Custom Architecture / Variability

Rocky 7

- Video Switcher
- VME Arch
- m68k Arch
- Framegrabbers
- Digital I/O
- Analog I/O
- Wireless ethernet

- Parallel Custom Interface
- MUX/Handshaking
- PID Controllers
- Actuator/Encoders

- Accels
- Gyros
- AIO

- Potentiometers
Supporting Different Platforms

*CLARAty adaptation process...*

Non-Reusable Layer

- ControlledMotor
- Joint
- Linear_Axis
- Mz<Type>

Non-Reusable Layer

- Fido_Motor
- R7_Motor
- R8_Motor
- Sim_Motor
- PID Controller
- DIO
  - MSI P430
- Analog IO
  - MSI P415/30
- LM629_Chip
- HCTL_Chip
- Widget_Motor
- R7_MC_Board
- Widget_Board

Legend:

- Non-Resuable
- Resuable
# Code Reusability

<table>
<thead>
<tr>
<th>Rocky 7 Modules</th>
<th>Lines of Code</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled Motor</td>
<td>2,652</td>
<td>Reusable</td>
</tr>
<tr>
<td>Input Output</td>
<td>2,690</td>
<td>Reusable</td>
</tr>
<tr>
<td>Bits</td>
<td>1,580</td>
<td>Reusable</td>
</tr>
<tr>
<td>Resources (Timers, etc.)</td>
<td>725</td>
<td>Reusable</td>
</tr>
<tr>
<td>Rocky 7 Motor</td>
<td>927</td>
<td>Non-usable</td>
</tr>
<tr>
<td>Rocky 7 H/W Maps</td>
<td>841</td>
<td>Non-usable</td>
</tr>
<tr>
<td>Motor Controller LM629</td>
<td>1,143</td>
<td>Reusable</td>
</tr>
<tr>
<td>Digital I/O Board (S720)</td>
<td>576</td>
<td>Reusable</td>
</tr>
<tr>
<td>PCI Components</td>
<td>329</td>
<td>Reusable</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>11,463</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Total Reusable</strong></td>
<td><strong>85%</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FIDO Modules</th>
<th>Lines of Code</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled Motor</td>
<td>2,652</td>
<td>Reusable</td>
</tr>
<tr>
<td>Trajectory Generator</td>
<td>691</td>
<td>Reusable</td>
</tr>
<tr>
<td>PID Controller</td>
<td>997</td>
<td>Reusable</td>
</tr>
<tr>
<td>Input Output</td>
<td>2,690</td>
<td>Reusable</td>
</tr>
<tr>
<td>Bits</td>
<td>1,580</td>
<td>Reusable</td>
</tr>
<tr>
<td>Resources (Timers, etc.)</td>
<td>725</td>
<td>Reusable</td>
</tr>
<tr>
<td>Common Definitions</td>
<td>2,380</td>
<td>Reusable</td>
</tr>
<tr>
<td>FIDO Motor</td>
<td>2,086</td>
<td>Non-reusable</td>
</tr>
<tr>
<td>FIDO H/W Maps</td>
<td>1,494</td>
<td>Non-reusable</td>
</tr>
<tr>
<td>Encoder Counter (ISA P 400)</td>
<td>463</td>
<td>Reusable - H/W</td>
</tr>
<tr>
<td>Analog Input Board (MSI P415)</td>
<td>519</td>
<td>Reusable - H/W</td>
</tr>
<tr>
<td>Analog Output Board (MSI P460)</td>
<td>462</td>
<td>Reusable - H/W</td>
</tr>
<tr>
<td>Digital I/O Board (MSI P560)</td>
<td>602</td>
<td>Reusable - HCTL</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>17,341</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Total Reusable</strong></td>
<td><strong>79%</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rocky 8 Modules</th>
<th>Lines of Code</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlled Motor</td>
<td>2,652</td>
<td></td>
</tr>
<tr>
<td>Trajectory Generator</td>
<td>691</td>
<td></td>
</tr>
<tr>
<td>Input Output</td>
<td>2,690</td>
<td></td>
</tr>
<tr>
<td>Bits</td>
<td>1,580</td>
<td></td>
</tr>
<tr>
<td>Resources (Timers, etc.)</td>
<td>725</td>
<td></td>
</tr>
<tr>
<td>Rocky 8 Motor</td>
<td>1,180</td>
<td></td>
</tr>
<tr>
<td>Rocky 8 H/W Maps</td>
<td>626</td>
<td></td>
</tr>
<tr>
<td>Widget Board Software</td>
<td>2,126</td>
<td></td>
</tr>
<tr>
<td>Motor Controller HCTL</td>
<td>900</td>
<td></td>
</tr>
<tr>
<td>I2C Master</td>
<td>1,165</td>
<td>Reusable - I2C</td>
</tr>
<tr>
<td>I2C Master Traci</td>
<td>1,223</td>
<td>Reusable - Traci</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>16,314</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Total Reusable</strong></td>
<td><strong>89%</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Total Reusable - Strict</strong></td>
<td><strong>56%</strong></td>
<td></td>
</tr>
</tbody>
</table>
Generic Algorithms

- Actual Example of Code Reusability for software modules:
  - Wheeled Locomotor Hierarchy – works for Rocky 8, Rocky 7, Fido, K9, and much more

(a) Skid Steering (no steering wheels)
(b) Tricycle (one steering wheel)
(c) Two-wheel steering
(d) Partially Steerable (e.g. Rocky 7)
(e) All wheel steering (e.g. Rocky8, Fido, K9)
Example:

**Collaborative Development for Locomotor**

**Version 1.0**
- Designed for Rocky 7
- Used Motor class
- Separated wheel control from locomotion
- Built-in pose estimation

JPL - 1998

**Redesign/mature**

**Version 2.0**
- Generalized design for wheeled locomotors
- Full and partially steerable vehicle
- Used generic motor classes
- Implements fixed axle model
- Developed continuous driving
- Adapted to Rocky 8, Rocky 7, and Sim

JPL - 2001

**Version 3.0**
- Separated model from control
- Add separate locomotor state
- Add concept of wheel and steerable wheel, Drive Cmd, Drive Sequence
- Adapt to ATRV, Sim, Rocky 7, Rocky 8

CMU - 2002

**Version 4.0**
- Use device and telemetry infrastructure
- Add adaptation to K9

In Progress
ARC - 2003
Multi User Integration
CLARAty Development Team

NASA Ames Research Center
- Maria Bualat
- Sal Desiano
- Clay Kunz (*Data Structure Lead*)
- Randy Sargent
- Anne Wright (*Cog-E and core lead*)

Carnegie Mellon University
- David Apelfaum
- Reid Simmons (*Navigation lead*)
- Chris Urmson
- David Wettergreen

University of Minnesota
- Stergios Roumeliotis
- Master Student

MIT
- Brian Williams
- Greg Sullivan

Jet Propulsion Laboratory
- Max Bajracharya (*Cog-E & vision lead*)
- Edward Barlow (34)
- Antonio Diaz Calderon (34)
- Caroline Chouinard (36)
- Gene Chalfant (34)
- Tara Estlin (36) (*Decision Layer lead*)
- Erann Gat (36)
- Dan Gaines (36) (*Estimation Lead*)
- Mehran Gangianpour (34)
- Won Soo Kim (34) (*Motion lead*)
- Michael Mossey (31)
- Issa A.D. Nesnas (34) (*Task Manager*)
- Richard Petras (34) (*Adaptation lead*)
- Marsette Vona (34)
- Barry Werger (34)

OphirTech
- Hari Das
About 300 modules in Repository
About 500,000 lines of C++ code
Five rover adaptations: R7, R8, FIDO, K9, ATRV
Most technology modules are at Level I and Level II integration
CLARAty Testbeds

- FIDO Benchtop
- Lab Webcam
- Rocky 8 Benchtop
- Dexter Manipulators Rocky 7 Benchtop
Levels of Integration

**Level I - Deposited**
- Code exists in CLARAty repository - all Intellectual Properties items cleared
- Compiles as a standalone application - no dependencies to other modules
- Have test programs and user documentation for getting started

**Level II - Encapsulated**
- Integrated with other CLARAty modules
- Uses CLARAty components to interact with rover
- Does not support a CLARAty API
- Runs on at least one robot platform

**Level III - Integrated**
- Conforms to a generic CLARAty API (or parent class)
- Has no unsupported 3rd party dependencies
- Runs on all applicable rover platforms

**Level IV - Refactored and Reviewed**
- Software reviewed by committee to ensure internal/external consistency
- Uses all applicable CLARAty classes
- Internally conforms to CLARAty conventions and coding standard

**Level + - Reused**
- Re-used by other modules in CLARAty - dependent module
- Provides access to all internal data products
Is CLARAty Paying Off?

Some Data Points

- After integrating and tuning EKF into new estimation framework on Rocky 8 (3 months), integrating algorithm for FIDO took **3 days**!
- After testing CLARAty/Morphin navigator on Rocky 8, integrating and testing on FIDO took **2 days** and on K9 **1 week**
- After getting locomotion working on FIDO, moving to K9 took **4 days**
- After getting mast software to work on Dexter, moving to Rocky 8 took **2 weeks** and to FIDO **4 days**
- Adaptation of entire locomotion, motion control, I/O control and communication onto completely new avionics FPGA hardware with PPC405 took **2 weeks**

N.B. These results are from professional CLARAty developers and should not be used to access development for new users
## Technology Algorithms in CLARAty

### Selected Robotics Algorithms (in CLARAty)

<table>
<thead>
<tr>
<th>Package</th>
<th>Algorithm</th>
<th>Integration Level</th>
<th>Provider</th>
<th>Supported Platforms</th>
<th>Tested Platforms</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation</td>
<td>Morphin</td>
<td>III</td>
<td>Simmons</td>
<td>All</td>
<td>R8</td>
<td>R8</td>
</tr>
<tr>
<td></td>
<td>GESTALT</td>
<td>II</td>
<td>Maimone</td>
<td>R8</td>
<td>R8</td>
<td>Converting points cl</td>
</tr>
<tr>
<td>Path Planning</td>
<td>Grid Mapper</td>
<td>III+</td>
<td>Urmson</td>
<td>All</td>
<td>R8</td>
<td>Not fully exercise</td>
</tr>
<tr>
<td></td>
<td>D*</td>
<td>II</td>
<td>Stentz</td>
<td>All</td>
<td>R6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tangent Graph</td>
<td>I+</td>
<td>Laubach</td>
<td>Sim</td>
<td>Sim</td>
<td></td>
</tr>
<tr>
<td>Pose Estimation</td>
<td>EKF</td>
<td>II+</td>
<td>Baumgartner</td>
<td>R8</td>
<td>R8</td>
<td>Migrating to new support all platfor</td>
</tr>
<tr>
<td></td>
<td>Visual Odometry</td>
<td>II+</td>
<td>Mathies, Cheng</td>
<td>All</td>
<td>FIDO, R8, R7</td>
<td>R7 (only older ver)</td>
</tr>
<tr>
<td>Vision</td>
<td>Wheel Odometry</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>FIDO, R8</td>
<td>Fixed Steering</td>
</tr>
<tr>
<td></td>
<td>Corner Detection</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>R8, R7, K9</td>
<td>Harris</td>
</tr>
<tr>
<td></td>
<td>Edge Detection</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>-</td>
<td>Canny</td>
</tr>
<tr>
<td></td>
<td>Image Rectification</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Image Pyramidizing</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stereo processing JPL</td>
<td>III+</td>
<td>MV - many</td>
<td>All</td>
<td>R8, R7</td>
<td>Binaries</td>
</tr>
<tr>
<td></td>
<td>Stereo processing SVS</td>
<td>II+</td>
<td>SRI</td>
<td>x86</td>
<td>x86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2D Feature Tracker</td>
<td>III</td>
<td>Bajacharya, Bandari</td>
<td>All</td>
<td>R8, R7</td>
<td>2D</td>
</tr>
<tr>
<td></td>
<td>2D/3D Feature Tracker</td>
<td>III</td>
<td>Nesnas et al</td>
<td>All</td>
<td>-</td>
<td>Expected Date:</td>
</tr>
<tr>
<td></td>
<td>Wide Baseline Stereo</td>
<td>II - III</td>
<td>Olson</td>
<td>All</td>
<td>-</td>
<td>Expected Date:</td>
</tr>
<tr>
<td>Locomotion</td>
<td>Wheel vehicles foward kinematics</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>R8, R7, FIDO, ATRV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wheel vehicles inverse kinematics</td>
<td>III+</td>
<td>Many</td>
<td>All</td>
<td>R8, R7, FIDO,</td>
<td></td>
</tr>
</tbody>
</table>
Competitive and Complementary Algorithm Integration
Architectural Traverse Example (1)

- **Rover**: K9 Rover
- **State Updates**: Commanding and State Updates
- **Path Information**: Path Planner
- **Navigator**: R7/Soj Navigator
- **Mapper**: Obstacle Mapper
- **Terrain Sensor**: Stereo Engine
- **Stereo Camera**: Camera R, Camera L
- **Locomotor**: K9_Locomotor
- **Pose Estimator**: EKF Pose Estimator
- **Stereo Processor**: JPL Stereo
- **Tangent Graph**:

Asynchronous:
- Rover: e.g. Rate Set at: 5 Hz
- Mapper: e.g. Rate Set at: 8 Hz
- Terrain Sensor: Synchronous/or Asynchronous e.g. Rate Set at: 10Hz used by other activities

Active:
- Rover
- Mapper
- Terrain Sensor
Technology Validation
Process for Mission Scenarios
'09 MSL Validation Scenario #1: Long Range Traverse

**Description:** Enable autonomous traverse, obstacle avoidance, and position estimation providing up to 100m/sol with less than 3% error relative to starting position.
'09 MSL Validation Scenario #2: Approach & Instrument Placement

Description: Enable placement of a science instrument on a designated target, specified in imagery taken from a stand-off distance. Placement accuracy to be within 1cm or 0.1%, from a stand off distance not greater than 10m.

Max designation range < 10m
'09 MSL Validation Scenario #3:
Onboard Science Data Processing

Description: Enable processing of science data onboard the rover system. This will be used for the progressively more challenging issues of:

- intelligent data compression (inlier detection) and prioritization,
- anomaly recognition (outlier detection) with stop and communicate result,

Example: "Terrain classification while doing long traverse"
Mission Infusion
2009 Proposed Rover Configuration

- UHF Helix Antenna
- Science Stereo Imager and Navcams
- IR Spectrometer
- Remote Sensing Mast
- Primary Arm with Surface Abrader, Corer and Microscope
- X-Band HGA (LGA not shown)
- Secondary Arm for Scoop and Raman Spectrometer Probe
- Payload Module containing Analytical instruments and SA/SPaH
- Surface Radiator (2x)
Technology Component Flow

1. NRA Proposals
2. NASA Selection
3. MTP Strategy
4. MER Robotics Technology
5. MTP Competed Robotics Technology
6. Other Robotics Technology Programs
7. Legacy Robotics Technology
8. MTP led Review Committee
9. MSL / MTP
10. MSL Technology Validation Tasks
11. CLARAty Task
12. MSL MDS Task
13. Other Projects
Overview of Software Flow in Time

- MSL Flight Software
- MSL Rover Testing
- Legacy S/W
- MTP Tech Dev
- MTP Tech Dev

Years:
- 2002
- NOW
- 2005
- 2009
Summary

- Previous point design of systems has yielded some solutions but lack critical mass amongst all NASA robotics efforts.

- Cross-leveraging and comparison of solutions requires extensive software infrastructure development, underway.

- A major product is validation for mission infusion.

- Another major product is building a legacy of robust, documented software and algorithms for future leveraging by technology development efforts.