Rover Technology Development and Infusion for the 2009 Mars Science Laboratory Mission

Richard Volpe and Stephen Peters
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91103
email: {firstname.lastname}@jpl.nasa.gov

Keywords: Spacecraft Autonomy, Planetary Robotics, Technology Infusion

Abstract

This paper provides an overview of the rover technology development, integration, validation, and mission infusion process now being used by the NASA Mars Technology Program. Described are the relevant mission scenarios of long range traverse and science instrument placement, along with the enabling algorithmic components for them. These are being integrated into the *CLARAty* robotics research software environment for demonstration and component validation, and then infused into the *MDS* spacecraft flight software environment for system level validation. As discussed, these components come from the ongoing 2003 rover mission, funded MTP research, and other complementary sources.

1. Introduction

The Mars Science Laboratory Project (MSL) is currently studying concepts for a landed mission to investigate the surface of Mars for the 2009 launch opportunity. The option for a mobile system is receiving particular attention. The surface mission duration under consideration would be significantly longer than that of the Mars Exploration Rovers (MER) to be launched in May and June of 2003: five hundred sols or more.

A Science Definition Team was formed to provide science direction for the mission, releasing its report in October of 2001 [1]. One significant characteristic of the recommended mission which distinguishes it from MER is the acquisition of rock and soil samples for analysis by laboratory science instruments on board the spacecraft. While a static option was also described, the mobile option was characterized by significant mobility, with a range over the life of the mission of possibly tens of kilometers, capable of taking the mobile laboratory from its touchdown point within the landing error ellipse to a predetermined planetary target, and capable of visiting multiple geological units separated by a few kilometers.

An increased rate of science return relative to MER could be enabled with increased robotic autonomy, reducing dependence upon ground-based engineering supervision of robotic actions. Improved high-level rover system control, including resource-cognizant task execution, would also enable a remote system to accomplish more between scheduled contacts with Earth. A number of applicable robotics and automation technologies have been, and are currently being, developed and demonstrated within research programs: these technologies need further maturation and/or validation in order to demonstrate their applicability and readiness for incorporation within the MSL mission software system, which is based upon the Mission Data System (MDS) architecture [2].

A technology program focused on maturing applicable technologies for MSL, and demonstrating that they can be integrated and infused into the mission flight software system, has been established to bridge the gap between research and flight readiness. All technology intended for MSL mission infusion will be demonstrated by the MSL Preliminary Design Review in August, 2005.

2. Rover Technology Infusion

As more surface missions are anticipated for Mars, with elevated expectations of mobility and autonomy, it becomes important to develop a process for capture of advanced research capabilities in flight systems. Up to the present, this has often been accomplished by having technology developers assume positions on the flight team, and bring their technology components with them. However, such a process is not always feasible or desirable, and is biased against technology developers not located at the institution of the flight project.

The Mars Technology Program has attempted to remedy this situation by developing a process by which technology providers infuse their component technologies into a coherent whole, where they may be leveraged by other participants, compared with competing techniques, and validated for capture by upcoming missions. This process is designed to be distinctly different than its predecessors in the way it organizes participants, captures their technology

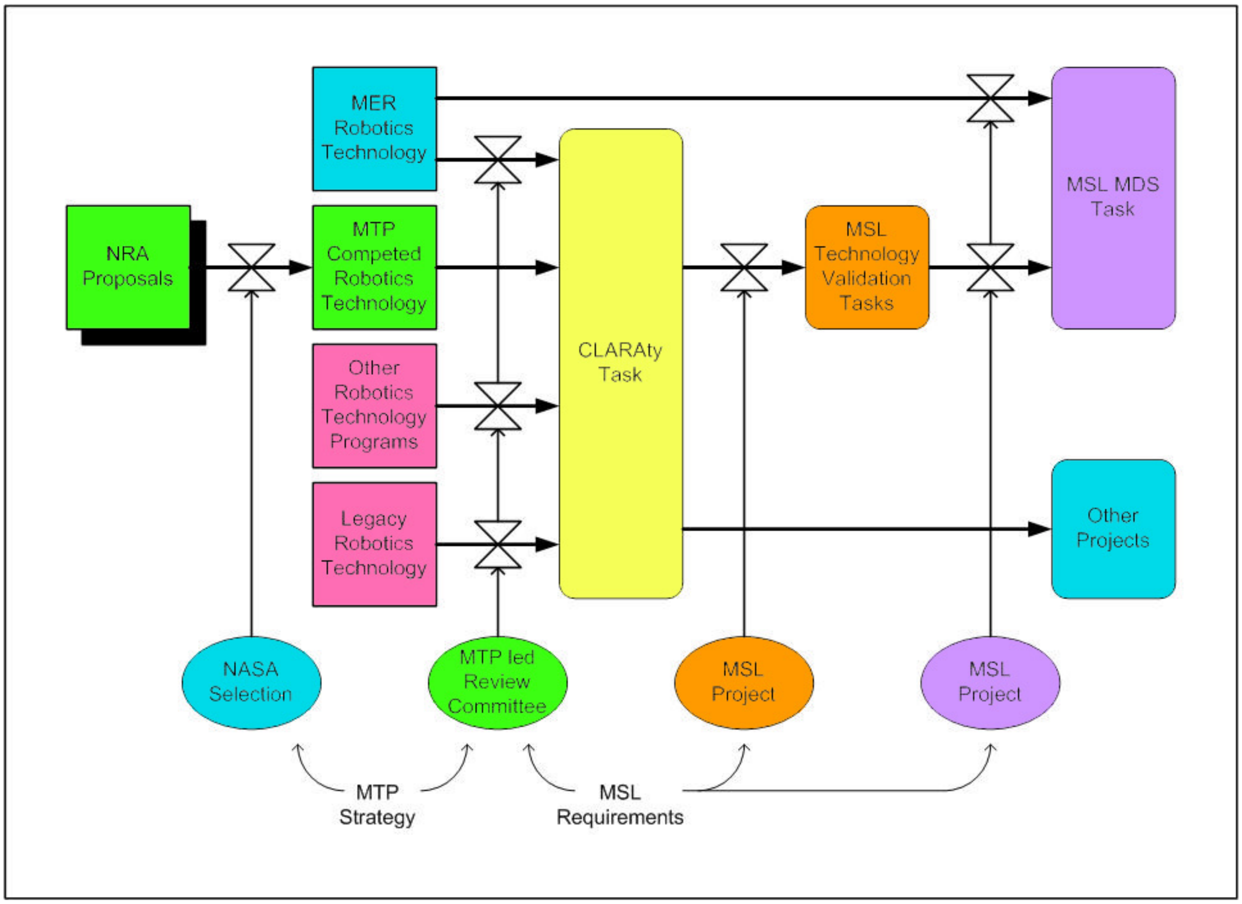

Figure 1 – Rover Functional Autonomy Technology Flow

products, and experimentally validates the resulting system capabilities prior to infusion into the mission. A diagram of the process flow is shown in Figure 1, and it will be described throughout the remainder of this paper.

First, all technology providers are competitively selected through proposal calls and technical evaluation [3]. The content of the call is based on specified mission needs, currently provided by MSL. Resultant proposals must demonstrate that the technology to be provided is reasonably mature, addresses mission needs, lives within mission constraints, and can be transferred to MTP within the period of funding. Maturity should be at Technology Readiness Level (TRL) four at the start of funding, and demonstrated in the integrated MTP system at level six by the conclusion of funding [4].

Second, the product of these efforts is not just journal papers and documented results. Rather, the primary product of all providers is software delivered to MTP by integration into a common software environment. This software system, the Coupled Layer Architecture for Robotic Autonomy or CLARAty [5], is being actively developed and provided by MTP, and its support team actively assists technology providers with integration of their software products. Further, the CLARAty infrastructure itself is being developed through multi-institutional collaboration between JPL, NASA Ames Research Center (ARC), and Carnegie Mellon University (CMU). CLARAty will be reviewed in Section 6.

Third, with the research products integrated into a common software system, they may be combined, compared, and quantified in their performance. Further, since CLARAty provides abstraction of, and support for, numerous test platforms, the performance may be elucidated independent of single platform particularities. Included amongst the platforms is a rover simulation system, ROAMS [6], which will allow for test trial repetition not possible by slower experimentation with physical rovers. Therefore, quantification of software and algorithm performance will not only be based on experimentation, but statistical results from simulation. Further, the

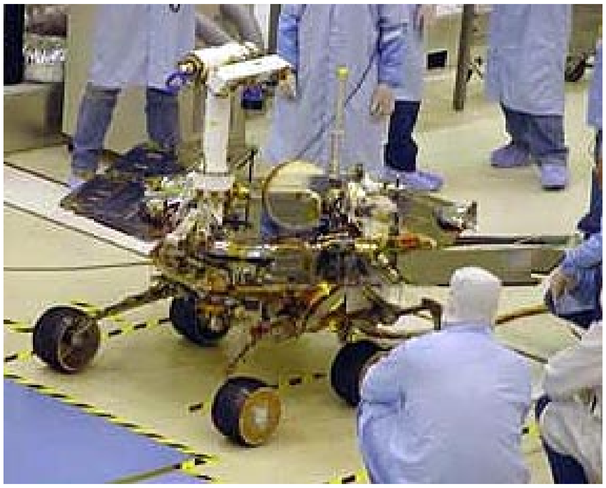

Figure 2 – One of the twin Mars Exploration

experimentation will validate the simulation fidelity. The documented results will quantify the performance of the individual technology products, as well as their integrated configurations, in mission relevant scenarios.

Fourth, based on the validation of the component technologies, the flight project may make well informed decisions about which subset of software products will be used in the missions. These items are then integrated into the Mission Data System (MDS) for full spacecraft system validation, in a phased and continuous process. Once in MDS, flight qualification for the Mission is directly possible after technology infusion is complete. MDS will be reviewed more in Section 8.

The following sections will describe in more detail examples of technology components going through this infusion process, the software architecture that binds the them together during development, the validation scenarios used in measuring their performance, and the mission software architecture into which they are finally captured.

3. 2003 Mars Exploration Rover Technology

In May and July of 2003, the MER twin rovers will be launched, arriving at Mars in early 2004. (Figure 2 shows one of these rovers being tested in late 2002.) Once safely reaching the surface, a number of new capabilities will be utilized to drive to science targets and place instruments against them. The robotic capabilities are the product of previous NASA funding in the research program, transferred to the mission through an inconsistent process of software infusion. But once validated and used by the mission, they necessarily become the de facto standard for future mission performance comparison. Therefore, it is important to migrate these into CLARAty so that new technology products can be directly compared against them.

There are several robotic capabilities that define the baseline onboard MER:

- Stereo vision [7,8]
- Obstacle Detection and Navigation [9,10]
- Vehicle Kinematics
- Position Estimation

In addition, there are several complementary off-board capabilities from MER that are intended for use either within, or in conjunction with CLARAty:

- Science Activity Planner [11]
- Rover Simulation [6]
- Camera and manipulator calibration [12]
- Motion Planning [13]

4. Mars Technology Program

The Mars Technology Program, in conjunction with the Mars Science Laboratory Mission, is funding three complementary infrastructure elements: ROAMS, WITS, and CLARAty. MTP is also funding eight competitively selected technology providers, and will be adding to this number through upcoming NASA Research Announcements (NRAs) [3]. Finally, MTP is funding the infusion of these technologies into MDS.

4.1 Software Infrastructure

Rover Analysis Modeling and Simulation Software, or ROAMS [6], is a high fidelity rover simulation environment built upon a Dynamics and Real-time Simulation engine (DARTS) which was the 1997 recipient of NASA software of the year. The same underlying DARTS software is used for Entry, Descent, and Landing Simulation, thereby providing a complete simulation system for MSL needs.

ROAMS provides simulation services for off-line analysis, as well as acting as a virtual rover platform for CLARAty control software. In the latter mode, actuators, sensors, and environment are simulated at different levels of resolution appropriate for the controls problem. For instance, if control of vehicle slippage is being tested, then simulation of wheel-soil interaction is required. In this case, the interface to ROAMS is done at the level of individual wheels. Alternatively, when planning and execution algorithms are tested using the simulator, then connectivity is performed at the vehicle level, and wheel-soil interactions need not be explicitly calculated, instead modeled statistically if at all. This flexibility matches the simulation to the level of fidelity needed for the problem being addressed. It also, allows for increases in the speed of simulation, permitting more testing of lower frequency system control loops.

The Web Interface for Tele-Science, or WITS [11], is a operations software environment for perusal of rover telemetry and construction of sequences for rover control. A subset of its capabilities is used for the MER Science Activity Planner (SAP). It has also been demonstrated to provide a goal specification interface to planning and scheduling systems such as CASPER [14]. CASPER, in turn, has served as the prototype Decision Layer for the CLARAty architecture. Further interfacing between CLARAty and WITS will occur in FY03, tying WITS to the CLARAty Functional Layer, both for execution and telemetry.

Both the Decision Level and Functional Level of CLARAty will be described in Section 6. MDS will be described more fully in Section 8.

4.2 Competitively Selected Rover Tech Components

As previously described, technology software algorithm developers have been competitively selected to address the needs of the MSL mission. The software products from these teams are being integrated into CLARAty for use with rovers and simulation surrogates, and access by the WITS operations interface. Currently, there are eight funded research teams, with more expected in the near future:

- Driving on Slopes, JPL/Caltech [15]
- Visual Servoing, JPL/Caltech [16,17]
- Autonomous Science, NASA Ames [18,19]
- Fault Diagnosis, NASA Ames [20]
- Vehicle Planning, CMU [21,22]
- Mapping, University of Washington [23]
- Terrain Estimation, MIT [24]
- Position Estimation, University of Michigan [25]

5. Legacy and Other Technology

The active flight and research software development described above represents only a subset of technology available for capture and use on future rover missions such as MSL. There has been over 15 years of autonomous rover research funded by NASA, and many of the products of that funding do not have software implementations available, or the implementations are in heterogeneous systems [26,27,28,29,30,31]. To enable quantified performance according to metrics, and qualified performance by comparison to competitive techniques, it is necessary to bring these legacy technology products into a common software environment. There are several issues to be considered when reviewing and prioritizing legacy technology products:

- applicability to currently planned missions
- overlap with currently funded or integrated products
- level of maturity previously achieved
- completeness and quality of documentation
- ease of software capture or re-creation

All of these factors translate into a cost/benefit ratio that must be developed and prioritized. Such an effort is currently underway through a recently formed inter-institutional team formed by MTP, and it is anticipated that initial efforts of capturing legacy products will begin in FY03.

In addition to legacy products, which by definition have no current funding base, there are also complementary technology products being developed in other programs. A case in point is the NASA Code R Intelligent Systems Program (IS). In the recent past, IS has largely concentrated on Decision Layer technology such as planning and scheduling. There has already been some progress in incorporating these results by interfacing to the resulting software, but the bulk of MTP efforts have concentrated on Functional Layer controls technology. More recently IS funding has begun to cover areas of control, making the projects very complementary to those of MTP. Also, maturing controls infrastructure in MTP has led to the desire to interface to more of IS Decision Layer products. Details of this interaction are still under development at the time of this writing.

6. CLARAty

The 'Coupled Layer Architecture for Robotic Autonomy', or CLARAty, has been developed to serve as the technology integration software architecture for MTP [5]. From the beginning, it has been designed to satisfy multiple objectives:

1. Provide a common software environment for heterogeneous rover research platforms, and transparently include simulated versions.
2. Provide a generalized, modular, and reusable software framework, that spans existing and past robotics research.
3. Provide tight coupling of the traditional artificial intelligence (AI) fields of planning, scheduling, and execution, with the traditional robotics fields of sensing, estimation, and control.
4. Satisfy the design and usage objectives of participating institutions, including JPL, ARC, and CMU.
5. Utilize contemporary development tools such as object-oriented programming, UML documentation, distributed and collaborative design and development, comprehensive version control, etc.

The resultant design is a dual layer architecture with a Decision Layer (DL) for AI software, and Functional

Layer (FL) for controls implementations. Implicit in the design is the concept of granularity, which increases within each layer. FL granularity allows for the nesting of capabilities and the hiding of system details, often through the use of polymorphism. DL granularity allows for variability in the planning system time quanta, and conditional goal expansion.

As described by Figure 1, CLARAty serves as the integration environment primarily for MTP funded research, but also for capture of MER flight capabilities, IS program software products, and other legacy software relevant to MSL. Through abstraction of the hardware layers, it currently enables these software products to be transparently used on 4 custom research rovers (Rocky 7, Rocky 8, K9, and FIDO), one commercial platform (ATRV), and benchtop duplicates of these systems' avionics.

Integration of technology products to the software architecture, instead of the individual platforms, has a number of advantages:

- Timesharing of platforms for development and testing.
- Experimental comparison of similar techniques on a single platform.
- Experimental demonstration of the robustness of a single algorithm on differing platforms.
- Distribution of parts of the whole rover control problem across multiple research teams, with integration of new products later into the whole.
- Leveraging of the integrated products of others by all teams, thereby reducing overhead and duplication of effort by all.
- Centralization of the final resultant software system, providing a single source of technology products for infusion to flight systems.

This final point provides a pathway to flight, but doesn't necessarily provide the needed information by which the flight project can properly select amongst all technology components available in the research software environment. Therefore, validation of each is needed to provide the information for this decision process, as described next.

7. Rover Technology Validation

After research technology products have been integrated into CLARAty and verified by the providers to perform as expected, there is still a need for additional extensive testing. This is to validate the technology, by using it with multiple rovers and numerous conditions, and quantifying its performance. There are a number of reasons for this need:

- To provide independent verification that the technology providers have delivered what was claimed, and quantify the performance.
- To provide possible feedback to technology providers enabling fixes or improvements of their products.
- To test single technology components interacting with each other, and confirm there are no algorithmic or architectural problems.
- To test combinations of technology components grouped to achieve a single mission designated capability.

Specific to this last item there are two primary mission capabilities designated by MSL: long traverse, instrument placement. In addition, research is also addressing the enhancing capability of autonomous science data processing.

Long traverse requires autonomously driving distances on the order of 100 times the vehicle length. Many terrain features of significance, such as obstacles, will typically not be apparent in panoramic imagery provided to operators by the rover from its starting location. High resolution imagery from orbit may help map large scale terrain qualities, and may be used by operators or onboard the rover for global path planning. However, determining the original position of the rover and maintaining an accurate estimate during the traverse become important issues. This is especially true in soft terrains which cause slippage, or featureless terrains where visual correlation is difficult.

Instrument placement requires approaching a terrain feature designated by scientists from up to 10 vehicle lengths distant, and reliably placing an instrument on the feature. An important facet of this capability is keeping track of the target even while traversing toward it through rough terrain. A continuous line of sight may not possible, and differences in lighting or view angle may complicate the process. Also, the rough terrain expected for rock fields of interest can make navigation and position estimation difficult. As the desired target becomes close to the vehicle, another complication may be introduced by the necessity to use cameras differing in focal length, stereo separation, field of view, and vehicle mount position. Finally, once the target is within the workspace of the manipulator with science instrument, the arm must be deployed safely and reliably to the target location. This last operation may require repetition for surface preparation steps, may require force control for grinding operations or surface compliance, and must handle contingencies through lighting changes and thermal cycles during long deployments.

Finally, autonomous science data processing is seen as a mission enhancing capability that will be extremely important during 500 day missions. Three types of data processing are possible:

Data Compression – This provides passive categorization, or compression of data collected for other purposes. Examples might be as simple as cropping sky from images taken for geology, or using navigation imagery to quantify rock distributions during traversal.

Activity Suspension – This requires detection of known features using periodic measurements, and aborting current plans if specified conditions are met. An example of this type of capability would be to monitor periodic spectral readings and abort the remainder of a traverse if a carbonate signature is detected.

Conditional Activity Initiation – This is similar to above, except rover activities are initiated without further review by ground operators. An example would be suspension of a long traverse and initiation of an instrument placement operation, based on data collected during the traverse. While this level of capability is a goal for the technology program, it is currently considered by many as too aggressive for MSL.

Currently two funded activities are in progress to perform validation for long traverse and instrument placement. It is planned that a third activity will address autonomous science data processing validation beginning in FY04.

Once technology components have been validated through this process, they may be selected for infusion in the flight software environment, MDS, described next.

8. Mission Data System

The Mission Data System is a new flight mission software architecture currently under development at the Jet Propulsion Laboratory. It combines a disciplined systems engineering approach, called "state analysis," with a corresponding software implementation architecture.

The design of MDS has been guided by a set of architectural themes [32] addressing needs ranging from software reliability and multiple mission re-use to support for the increased autonomy required by more ambitious future missions. MDS provides a common architecture for ground, simulation, and flight software elements which simplifies system design and facilitates incremental migration of capability from ground to flight software.

State-Based Architecture

Central to MDS is the concept of state: state of the spacecraft such as the temperature of a motor or the configuration of a manipulator, and state of the environment such as the location of obstacles to rover mobility or the direction of Earth. The evolution of these states over time is determined by the physics of the system and the behavior of the spacecraft as it responds to commands and sensor data.

All control within the system is expressed in the form of constraints on state, which are called "goals." Complex activities are expressed as goal networks, which contain goals and temporal constraints upon their start and end times.

State Analysis

The process of state analysis uncovers the set of states within the system: how they interact, and how they are monitored and controlled. This analysis maps directly to a set of software system building blocks, the architectural elements, from which the system is constructed [33]. Domain-specific algorithms such as rover navigation and manipulation are mapped to, and software implementing them distributed among, appropriate architectural elements.

Mission Planning and Execution

High-level overall control of the system is managed by the Mission Planning and Execution (MPE) portion of MDS. A high-level goal net representing the activity planned for the day is uplinked to the spacecraft. Goals expressing ultimate intent, such as having an instrument in place on a target rock, are expanded or "elaborated" into networks of sub-goals, such as maintaining the temperature of the actuators within operational limits. These elaborations will have been defined as part of the state analysis process.

Actual uplink is in the form of Goal Elaboration Language (GEL) statements, which can define not only goal networks, but also goal elaborations. Upon receipt of the uplinked goal net, MPE performs any necessary elaborations and merges the result with the currently executing goal net, making scheduling decisions as necessary using the flexibility available within the temporal constraints.

The degree of on-board flexibility in execution can be controlled from the ground by choosing how much elaboration is done on the ground prior to uplink and how much is left to be done on board the spacecraft.

Fault monitoring and response is implemented using the same goal net mechanism as that for nominal operations.

9. Integrated Technology Demonstration

The Mars Science Laboratory project has selected MDS as the baseline for its mission software. Because MSL is to be the first mission to use MDS, the MSL focused technology development program includes demonstration of the readiness of MDS itself to meet MSL's needs. An integrated technology demonstration has been planned with objectives to: 1) demonstrate that MDS is mature and ready for use as mission software, 2) demonstrate that robotics algorithms with complexity of behavior desired by MSL can be implemented within MDS, and 3) provide the mission with a head start toward actual mission software. The demonstration will include long distance traverse and single-sol target approach and instrument placement.

Incremental Implementation

MDS has a well-defined development methodology which includes rigorous configuration control and management of work flow. In order to avoid big-bang integration and associated problems, software is developed in small increments which are integrated into, and tested with, the entire system as they are implemented.

Planning takes place on two levels. A series of quarterly releases with associated new capabilities is defined. Then, each release is broken down into a set of increments of a few days of work each. Dependency analysis is used to phase and schedule the work.

The first step for each increment is state analysis, where details of the design, including data representation and interfaces between architectural elements, are documented and reviewed.

Implementation follows the documentation. As each increment is implemented, it is integrated into the complete system, its unit tests are added to the system test suite, and all system tests are run on the new build. This ensures that the new software doesn't "break" previously implemented capabilities and that the system functions as a coherent whole.

Phased Technology Infusion

Robotics algorithms contributing to long distance traverse and single-sol instrument capabilities are identified and allocated to the quarterly MDS releases, with the more mature algorithms implemented earlier. The algorithms included in the integrated demonstration may not be the final set of algorithms actually selected by MSL for flight, but the intent is that they comprise a sufficient set. Technology providers are expected to participate in the state analysis of their algorithms and assist in the infusion process.

Status

In August, 2002, the first demonstration of MDS running on board a rover was performed. Controlled by an executing goal net, and using encoder and potentiometer feedback, Rocky 7 performed a sequence of driving straight and turning in place. In February, 2003, control of a stereo hazard camera pair, stereo image processing, and the MER GESTALT navigation software [9] were added and demonstrated, controlled by uplink of a goal net expressed in GEL.

Enhanced vehicle position estimation is planned for the next demonstration, in August, 2003.

Acknowledgement

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] R. Arvidson, et al., “NASA Mars Exploration Program Mars 2007 Smart Lander Mission Science Definition Report,” 11 October 2001.

[2] S. Krasner, "A Reusable State-Based Guidance, Control and Navigation Architecture for Planetary Missions," Proceedings of 2000 IEEE Aerospace Conference, March 2000.

[3] http://research.hq.nasa.gov/

[4] http://www.hq.nasa.gov/office/codeq/trl/

[5] R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "The CLARAty Architecture for Robotic Autonomy," proceedings of the 2001 IEEE Aerospace Conference, Big Sky Montana, March 10-17 2001.

[6] J. Yen and A. Jain, "ROAMS: Rover Analysis Modeling and Simulation Software," in International Symposium on Artificial Intelligence, Robotics and Automation in Space
(i-SAIRAS), Noordwijk, Netherlands, June ‘99.

[7] L. Matthies, ``Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-time Implementation", International Journal of Computer Vision, 8(1), July 1992.

[8] Y. Xiong and L. Matthies, "Error Analysis of a Real-Time Stereo System", IEEE Conference on Computer Vision and Patter Recognition (CVPR), Puerto Rico, June 17-19, 1997.

[9] S. Goldberg, M. Maimone, and L. Matthies, “Stereo Vision and Rover Navigation Software for Planetary Exploration,” 2002 IEEE Aerospace Conference, Big Sky Montana, March 2002, pp 2025-2036.

[10] S. Moorehead, R. Simmons, D. Apostolopoulos, and W.L. Whittaker, “Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica,” International Symposium on Artificial Intelligence, Robotics and Automation in Space, June, 1999.

[11] P. Backes, K. Tso, J. Norris, G. Tharp J. Slostad, R. Bonitz, and K. Ali, “Group Collaboration for Mars

Rover Mission Operations," IEEE International Conference on Robotics and Automation, Washington DC, May 2002.

[12] D. Gennery, "Least-squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points," in Calibration and Orientation of Cameras in Computer Vision, A. Gruen and T.S. Huang, Editors, Springer Series in Information Sciences, Vol. 34, Springer-Verlag, pp. 123–136, July 2001.

[13] C. Leger. "Efficient Sensor/Model Based On-Line Collision Detection for Planetary Manipulators," IEEE International Conference on Robotics and Automation, Washington DC, May 2002.

[14] T. Estlin, R. Volpe, I.A.D. Nesnas, D. Mutz, F. Fisher, B. Engelhardt, S. Chien, "Decision-Making in a Robotic Architecture for Autonomy," The 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), Montreal Canada, June 18-21 2001.

[15] C. Olson, L. Matthies, M. Schoppers, M. Maimone, "Stereo Ego-motion Improvements for Robust Rover Navigation," IEEE International Conference on Robotics and Automation, Seoul Korea, May 2001, pp 1099-1104.

[16] M. Maimone, I.A.D. Nesnas, H. Das, "Autonomous Rock Tracking and Acquisition from a Mars Rover," International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), Noordwijk, The Netherlands, June 1999, pp. 329-334.

[17] D. Wettergreen, H. Thomas, M. Bualat, "Initial Results From Vision-based Control of the Ames Marsokhod Rover," IEEE Int'l Conf on Intelligent Robots and Systems, France, 1997, pp. 1377-1382.

[18] P. Gazis and T. Roush, "Autonomous identification of carbonates using near-IR reflectance spectroscopy during the February 1999 Marsokhod field tests," J. Geophys Res.-Planets, February 2001.

[19] V. Gulick, R. Morris, M. Ruzon, and T. Roush, "Autonomous Image Analyses During the 1999 Marsokhod Rover Field Test," Journal of Geophysical Research--Planets, 106(E4), 25 April 2001, pp. 7745-7764.

[20] V. Verma, R. Simmons, D. Clancy, and R. Dearden, "An Algorithm for Non-Parametric Fault Identification," AAAI Spring Symposium on Robust Autonomy, Stanford CA, March 2001.

[21] Stentz, A., "Optimal and Efficient Path Planning for Partially-Known Environments," IEEE International Conference on Robotics and Automation, May 1994.

[22] P. Tompkins, A. Stentz, and W. Whittaker, "Automated Surface Mission Planning Considering Terrain, Shadows, Resources and Time," The 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), Montreal Canada, June 18-21 2001.

[23] C. Olson, F. Xu, K. Di, R. Li, and L. Matthies, "Automatic Feature Registration and DEM Generation for Martian Surface Mapping," in proceedings of the ISPRS Commission II Symposium, August 2002.

[24] K. Iagnemma, H. Shibly, and S. Dubowsky, "On-Line Terrain Parameter Estimation for Planetary Rovers," IEEE International Conference on Robotics and Automation, Washington DC, May 2002.

[25] L. Ojeda, and J. Borenstein, "FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain," IEEE International Conference on Robotics and Automation, Washington DC, May 2002.

[26] "Robotic Vehicles for Planetary Exploration," IEEE International Conference on Robotics and Automation, Nice, France-May 1992.

[27] R. Volpe, J. Balaram, T. Ohm, R. Ivlev. "The Rocky 7 Mars Rover Prototype," IEEE International Conference on Intelligent Robots and Systems (IROS), November 4-8 1996, Osaka Japan.

[28] P. Schenker, et al., "FIDO: a Field Integrated Design & Operations Rover for Mars Surface Exploration," The 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS), Montreal Canada, June 18-21 2001.

[29] D. Christian, D. Wettergreen, M Bualat, K. Schwehr, D. Tucker, E. Zbinden, "Field Experiments with the Ames Marsokhod Rover," International Conference on Field and Service Robotics, Canberra Australia, 7-10 Dec 1997.

[30] E. Krotkov and R. Simmons, "Performance of a six-legged planetary rover: power, positioning, and autonomous walking," IEEE International Conference on Robotics and Automation, May 1992, pp. 169-174.

[31] D. Wettergreen, D. Bapna, M. Maimone, and H. Thomas, "Developing Nomad for Robotic Exploration of the Atacama Desert," Robotics and Autonomous Systems, Vol. 26, No. 2-3, February, 1999, pp. 127-148.

[32] D. Dvorak, R. Rasmussen, G. Reeves, A. Sacks, "Software Architecture Themes in JPL's Mission Data System," Proceedings of 2000 IEEE Aerospace Conference, March 2000.

[33] D. Dvorak, R. Rasmussen, T. Starbird, "State Knowledge Representation in the Mission Data System, Proceedings of 2001 IEEE Aerospace Conference, March 2001.