The Selection and Infusion of Autonomy for Mars Rovers

David F. Woerner

Abstract—**How do you select autonomy, insure that it meets your needs, and arrives for launch in a reliable form? How do you ensure autonomy providers around the country meet these goals compliant with institutional practices and policies at the Jet Propulsion Laboratory for building space qualified reliable systems? Answers to these questions have resulted in a process we call autonomy infusion. This paper describes the process MSL is using to infuse autonomy into a rover, and describes attributes, and evaluation criteria and their use pertinent to autonomy technologies for Mars rovers in general.**

Index Terms—**autonomy, rovers, Mars, reliability**

I. Introduction

IN the fall of 2009, National Aeronautics and Space Administration (NASA) will launch a mission to Mars carrying a rover that will conduct science experiments for an estimated 1000 days (~2.75 years). The number and complexity of those experiments is high when contrasted with Sojourner (Mars Pathfinder's rover, landed in 1997) and MER (Mars Exploration Rover, to land in 2004) [1]. Analyses indicate that many on-board operations must be made autonomous, and an intelligent system is baselined for this mission. Autonomy technologies enable this mission.

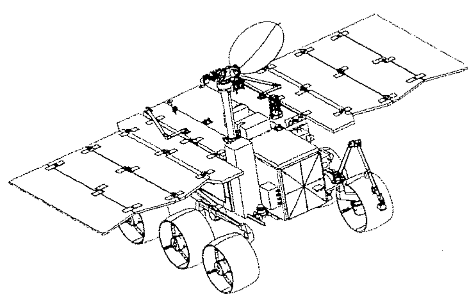

Fig. 1. Concept for the Mars Smart Lander rover.

Engineers and scientists at NASA have attempted to select, develop, and infuse autonomy within space missions many times. Most have been done in an ad hoc or experimental way. Requirements and objectives for the Mars Smart Lander (MSL) mission are sufficient to mandate autonomous operations on-board the vehicle for daily operations [1] and require a more disciplined approach to autonomy infusion. MSL concluded early on that a system of selection, development, and test would have to be created to infuse an intelligent and autonomous system into the MSL. The procedure would have to:

- guarantee the safety of the vehicle [2],
- produce complementary software and hardware designs that can not harm one another,
- provide a stable platform from which to conduct science,
- and significantly boost the performance of the vehicle over one not employing autonomy.

How do you select autonomy, insure that it meets your needs, and arrives for launch in a reliable form? How do you ensure autonomy providers around the country meet these goals compliant with institutional practices and policies at the Jet Propulsion Laboratory (JPL) for building space qualified reliable systems? Answers to these questions have resulted in a process we call autonomy infusion. This paper describes the process MSL is using to infuse autonomy into a rover, and describes attributes, and evaluation criteria and their use pertinent to autonomy technologies for Mars rovers in general

A. Surface Mission Description

The Mars Smart Lander Mission will feature a precision landing capability to get to within approximately 5 km of a given landing site [3]. This capability will allow landing in any of a large number of relatively safe places that are in close proximity to rougher areas of very high scientific interest. In addition, a terminal hazard detection and avoidance system will be used to select safe areas within the landing ellipse, thereby allowing a landing to occur in more hazardous terrain than previous missions. These capabilities in will allow delivery of approximately 1620 kg (820 kg for landing systems; 800 kg for surface systems, including approximately 70 to 100 kg of science payload) of landed assets in 2009 with a much wider selection of locations than possible, for example, with the 2003 Mars Exploration Rover (MER) landing system. This is the system for which autonomy technologies must be evaluated and selected.

B. Autonomy Infusion

At its most abstract, technology infusion revolves around three parameters: technology criticality, technology maturity,

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.

D. F. Woerner is with the Jet Propulsion Laboratory, Mail Stop 301-422, 4800 Oak Grove Drive, Pasadena, CA 91109 USA (telephone: 818-393-2000, email: david.f.woerner@jpl.nasa.gov).

and software maturity. Technology criticality measures the importance of a technology against the mission requirements and is a qualitative assessment that will change as the autonomy architecture matures. The maturity of a technology measures the degree to which a technology has moved from theory and into a usable product whose strengths and weaknesses have been characterized. The more times the technology has been applied and its quality recorded:

- then the more mature the technology is [4],
- the greater our understanding of the strengths and weakness of the technology [4],
- the appropriate use of the technology,
- and how the technology is best integrated and applied.
-

Technology maturity signals the confidence an engineer can have that a technology will meet the requirements. Software maturity measures the completeness, correctness, and reliability of software code that implements a technology. While they are certainly related, there is a distinction between the maturity of software and a technology that is implemented in the software. Software artifacts (the code and documentation) allow measurements to be made for a range of "ilities" such as reliability, stability, maintainability, reusability, etc.

Another factor contributing to the complexity of autonomy infusion is the coordination and development of technologies from many sources. JPL is responsible for the overall performance of the Mars Smart Lander s rover and science mission. MSL wants to use and rely on technologies JPL cannot provide alone. It is important to remember that technology maturation and validation is not the end objective. The ultimate objective is to operate a rover on Mars that meets the MSL mission requirements and maximizes the science return for one of NASA's most important missions.

Professors at institutions, researchers at laboratories, roboticists at JPL and NASA, and programmers in industry are and have resolved, or will resolve, some of the NP-hard problems posed by the MSL requirements. Yet these solutions are likely to have been arrived at in a research environment. These solutions may work in a terrestrial setting but have never been tried on Mars, or in space, on a large and complex vehicle working under constraints to complete its mission on time and on budget. Selection and infusion of these technologies cannot be an end in itself — ultimately MSL will be judged on its science return. Autonomy infusion was conceived to deal with the issues raised above and more.

The effort to infuse autonomy has been broken into 5 complementary tasks.

1. Identify a candidate set of mission-relevant autonomy technologies and providers across the nation including JPL/NASA, academia and industry.
2. Develop a set of criteria for evaluating the likelihood that candidate autonomy technologies can be matured and tested successfully by June 2005 and that can be used for selection of autonomy technologies.
3. Establish a testing methodology for candidate autonomy technologies which shall include the use of simulations and field tests, and provisions for system-level stress testing
4. Develop specific concepts for system-level validation of selected and integrated autonomy capabilities for the flight system.
5. Develop and publish guidelines to technology providers for assuring compatibility of candidate autonomy technologies with the Mission Data System (MDS) architecture. The Mission Data System is the baseline architecture selected for the MSL. Ergo, the autonomy providers have to design and develop their products to be compatible with the MDS. A description of with this means is beyond the scope of this paper.

While all 5 tasks are worthy of a detailed discussion, the focus of this paper is task number 2. The criticality of a technology, its maturity, and the maturity of the software implementation are all measured and ranked within this task

II. Critical Technology

There are many driving requirements for which critical technology must be developed for the MSL rover. The top ones (most costly, most complex to respond to, etc.) are:

- Long-range traverse, 10 to 50 km
- On-board planning and resource management
- In-situ science
- Mission Operations

Long range traverse sounds simple. The requirement is 450 m/sol (m/martian day). However this must be done in the blind since the mission operations team has no time to joy stick the rover, the rover must use its sensors to plan a path around rocks, gullies, and other obstacles to find science targets selected by the science team.

State of the art is 50m/sol, and this is what the MER rovers hope to achieve. Four hundred and fifty meters per sol will tax the MSL sensing system, its algorithms, and the nerves of the operations team. The autonomy technologies must not be easy to confound. This autonomy technology must sense and correctly interpret hazards to avoid and select a safe route to navigate without assistance.

The next requirement that drives autonomy is onboard planning and resource management. The rover must sense its state of health while in the Martian environment (for which the models are coarse at best) and plan its activities and consumption of limited resources to achieve its goals, again in a safe and timely manner without assistance.

The combination of states of the rover health and the world around it, and the desired state of the rover, as specified by the goals uplinked by the operations team, is a large and highly complex space from which to compute and execute acceptable solutions. Rover health and the state of the Martian environment are dynamic and consequently compound this space. Plans and resource management must continually be refined and checked to capture and respond to variances on the fly.

In-situ science poses some of the most severe challenges to the selection of rover autonomy technologies. MSL intends to rely on a suite of instruments that cover the electromagnetic spectrum from visible down to radio wavelengths, and carry a rock and regolith sampling system.

Instruments must be place and pointed with accuracy so that the science team receives data from targets they select. Errors in placement or pointing can result in false signals or at least wasted time in re-targeting and re-pointing sensors and sample collection systems [5]. The Earth will not be in contact with the rover to ensure the success of collecting in-situ science.

The final driving requirement discussed here is the one for operations of the autonomous flight system. Mission operations systems for planetary rovers have not traditionally relied on autonomy; their design is based on the flight systems using a very simple software system. It was perceived as too large a risk to put autonomy technologies into flight software. Consequently, ground systems have evolved and grown to operate simple software systems. The MSL operations system will have to deal with a rover that is making decisions without ground control or monitoring.

Mission operations systems will have to maximize the return of science while minimizing errors, personnel, and wasted time. The MSL operations software and processes must make decisions that do not harm the rover and result in "good" uplinks being sent to the vehicle.

III. Technology and Software Maturity

The selection of autonomy technologies is based on two similar sounding but very different precepts; one, the most reliable technologies will be the most mature ones, and two, the most mature software implementation of a technology will be the most reliable. Technologists offering maturity against both precepts will have an advantage during the initial evaluation process.

The two precepts are, however, too coarse to use for infusion of technologies into a Mars rover. They have been further refined into a set of general attributes that a technology and its implementation must have. These are loosely based on B. Boehm s work published in the paper, "Identifying Quality-Requirement Conflicts" [6]-[8].

A. Attributes

The attributes are captured in table I, where each attribute is labeled as applicable to technology maturity or software maturity or both.

TABLE I
ATTRIBUTES

Attribute	Technology Maturity	Software Maturity
Performance	X	X
Cost and Schedule	X	X
Simplicity	X	X
Assurance		X
Interoperability	X	X
Deployment		X
Usability	X	X
Maintainability		X
Reusability/Adaptability		X

These attributes offer wide latitude for interpretation. They could be applied to an algorithm, an architecture, or an implementation. As an example, "simplicity", could measure the elegance of an algorithm or the elegance of the instantiation of the algorithm in flight software. It is conceivable that an algorithm is simple in its published form, but when instantiated in software becomes convoluted and confusing. The inverse may also be possible. Which represents the preferred simplicity for the selection of technologies? This report makes the distinction clear in the upcoming sections.

The breadth of attributes make it clear that no single autonomy technology will excel in all attributes. It is even likely that entries responding to the same requirements will cluster near each other as measured for the same attribute. Final selections of autonomy technologies will require diligent and impartial examination of the entrants. The final selection team(s) will be composed of operational users, scientists, engineers, and project personnel as a minimum.

In addition, no one attribute can be used in isolation. A technology must score highly under several attributes to function as needed for MSL. Trades will have to be made as to what is an acceptable "basket" of attributes. This is likely to be a qualitative evaluation because extenuating circumstances will arise, requiring some qualitative judgements.

The rankings of the attributes is shown in table II.

TABLE II
ATTRIBUTE RANKING

Performance Cost and Schedule Simplicity	Highest
Assurance Interoperability Deployment	Mid-range
Usability Maintainability Reusability/Adaptability	Lowest

Attributes are measured using evaluation criteria. Generic criteria are captured in the following sections. Criteria specific to an algorithms or classes of algorithms may have to be developed by the project as the technologies make themselves known. It is not possible to estimate the number and variety of autonomy technologies a rover may use. Some criteria can be applied across the spectrum of autonomy technologies a mission may need or receive proposals for. Others may only apply to a type of technology. The project will measure "Return-On-Investment" by evaluating requirements compliance, employing the evaluation criteria outlined here, and then forming an opinion to select a proffered technology or not.

B. Evaluation Criteria

The evaluation criteria for each attribute were scrutinized until they were reduced to a reasonable set--deployable in relatively brief proposal reviews or walkthroughs of demonstration results. They were conceived from an interplanetary mission experience base and have not been generalized to broader applications. And the criteria are applied at two points in the infusion process, initial evaluation and final selection. Initial evaluation and final selection are separated by approximately 2 years.

Only 3 of the mentioned attributes are discussed in detail with their associated evaluation criteria due to space limitations in this paper.

1) Performance

The attribute performance led to criteria to be used to analyze, measure, and optimize selecting technologies that consume limited resources be it CPU utilization or memory; some of the more obvious ones. It also led us to such performance issues as required operations time to use (a technology) or "instrument placements" per command cycle. This attribute favors autonomy technologies that are faster, use less of a resource, and enhance the mission's ability to meet or exceed a requirement.

Modeling and performance prototyping should be used if possible at the initial evaluation and final selection; simulation, user involvement, and demonstrations are added at "final selection". Modeling and/or performance analysis prototyping will be required to support data provided with each proposal and the deliverables for final selection. Simulation and user involvement will be required for activities leading to a demonstration during final selection. A comparative analysis will be a natural fallout of the demonstration if more than one autonomy technology is available for a requirement; a set of comparative criteria has not yet been developed.

All of the selected technologies must fit within the limited resources available on the rover. The use of those resources will be measured and tallied. This system analysis may result in disqualification of a technology even though it passes the performance criteria.

TABLE III

EVALUATION CRITERIA FOR THE ATTRIBUTE PERFORMANCE

Criteria	Initial evaluation	Final selection
CPU usage	estimated	measured
Memory utilization (runtime and permanent storage)	estimated	measured
Uplink data volume required for routine servicing	estimated	measured
Amount of useful science return for given situation, including anomaly conditions	estimated	measured
Turnaround time to develop plan for given mission operations and conditions	estimated	measured
Turnaround time for execution of plan	estimated	
Number of goals achieved in given time	estimated	measured
Comparative performance results, if available		measured
Probability of outcomes for given situation: loss of single command cycle (enough info to recover)		measured
Probability of outcomes for given situation: loss of multiple cycles (requiring commanded diagnostics)		measured
Probability of exceeding rover resource limits		measured

estimated = estimated by technologist based on analytical methods, prototypes, or test results

measured = to estimate or appraise by criterion; a measurement is made to test against the criteria; this could be statistical, test results, etc. as is appropriate to the criteria; performed by JPL

2) Cost and Schedule

The need to review costs and schedule against the following criteria is invariant regardless of scope of work.

The costing of the development effort should address the life-cycle activities needed to:

- Develop requirements
- Translate requirements into designs
- Implement the designs and unit test the code
- Perform integration and pre-acceptance testing, including testbeds
- Correct defects and implement design changes
- Deliver and install the product, and train both users and operators

The schedule documents development plans, activities, and responsibilities at a level of detail that identifies the required resources and supports the monitoring of progress, the allocation of resources, the management of risk, and the attainment of the desired level of product quality.

At initial evaluation, a cost-to-complete and schedule will be required along with other items called out in this report [9], [10].

At final selection, an updated cost-to-complete and schedule will be required along with the deliverables for the demonstration on the rover. It is recommended that the evaluation team performing the final selection along with a technologist from the institution providing a technology, performs a risk assessment as follows:

- What is the risk in the cost estimate to completion? What is its source?
- What is the risk in the schedule estimate to completion? What is its source?
- How do cost and schedule risk relate to capability?
- How do cost and schedule risk relate to quality?
- Is the cost and schedule risk acceptable?
- How can the cost and schedule risk be mitigated?

TABLE IV
EVALUATION CRITERIA FOR THE ATTRIBUTE COST AND SCHEDULE

Criteria	Initial Evaluation	Final Selection
Is the schedule compatible with the project's autonomy infusion schedule?	measured	measured
Are the documentation deliveries included in the development schedule?	plan	
Is the cost-to-complete estimate acceptable?	measured	measured
What is the provider s track record for estimating cost and schedule for developing and deploying technologies (in general)? (accuracy and total experience)	documented	
Does the provider rely on processes oriented toward reuse?	plan	measured
Does the provider rely on automated processes	plan	measured
What is the provider s institutional track record for estimating cost and schedule for developing and deploying technologies (in general)? (accuracy and total experience)	documented	
Does the institution rely on processes oriented toward reuse?	documented	
Does the institution rely on automated processes	documented	
How extensive is the track record of the provider or institution of flight experiments or ops deployment?	documented	
What is offered by the provider institution as the basis of estimate? Is it credible? What is the probability of completing the design and maturation within the schedule and cost?	documented	measured
Is institutional infrastructure sufficient to mature technology, i.e., programmers and equipment	documented	documented
What is the caliber of personnel engaged in this development?	documented	documented
Has the provider been involved with the MSL team at an appropriate level?		substantiated
Have schedule and cost been tracked throughout technology maturation?	plan	substantiated
Does the development plan adequately document risks associated with this development?	documented	documented
Does the development plan adequately mitigate the risks associated with this development?	documented	documented

measured = to estimate or appraise by criterion; a measurement is made to test against the criteria; this could be statistical, test results, etc. as is appropriate to the criteria; performed by JPL
plan = a plan is called for and delivered to JPL
documented = response to criteria is documented and delivered to JPL
substantiated = to establish by proof or competent evidence; checked against criteria using observations, developer polls, document reviews, etc..

3) Simplicity

Simplicity is an attribute used to assess whether a technology is tractable along many dimensions. The associated evaluation criteria (table V) analyze, measure, and optimize selecting technologies that rely on the simplest interfaces, hardware design, and internal structures. Simplicity in an algorithm may be of value, and clearly simplicity in instantiation of an algorithm is of a high priority since it leads to flight software that is maintainable, testable, reusable, and supports interoperability.

TABLE V
EVALUATION CRITERIA FOR THE ATTRIBUTE SIMPLICITY

Criteria	Initial Evaluation	Final Selection
Number of layers (projected for autonomy technology delivery)	estimated	measured
Number of function points (projected for autonomy technology delivery)	estimated	measured
Uses code for invariants and data for specifics?	documented	documented
Has proper use of programming techniques been made (modules, class encapsulation, inheritance, and libraries)	plan	Documented/ substantiated
Classes of heuristics or heuristics used (projected for autonomy technology delivery)	documented	
Number of fuzzy parameters (e.g. decision thresholds projected in delivered autonomy)	estimated	measured
Number of interfaces with rover hardware and MDS	estimated	measured
Cost/mass/power for proposed hardware requirements (e.g. sensors)	estimated	measured
Impact on operations, i.e., uplink preparation time, downlink analysis time	estimated	measured
Is the technology too complex as implemented (too complex to read, easily understand, integrate, test and/or use)?		substantiated
Ease of review by, training of, new developer or user		substantiated
Lines of code (projected for autonomy technology delivery)	estimated	measured

estimated = estimated by technologist based on analytical methods, prototypes, or test results
measured = to estimate or appraise by criterion; a measurement is made to test against the criteria; this could be statistical, test results, etc. as is appropriate to the criteria; performed by JPL
plan = a plan is called for and delivered to JPL
documented = response to criteria is documented and delivered to JPL
substantiated = to establish by proof or competent evidence; checked against criteria using observations, developer polls, document reviews, etc..

IV. CONCLUSION

The MSL mission will systematically select and infuse autonomy technologies into its rover design. The process is described here and should be more reliable and effective than attempts on previous missions. The process does not evaluate autonomy based on specifics about the Mars Smart Lander and hence is extensible to Mars rovers in general.

V. ACKNOWLEDGMENT

David Woerner thanks Dan Clancy, Jim Murphy, Rich Washington, Rich Doyle, Steve Peters, Jack Morrison, Terry Huntsberger, and Rajendra Parikh for their efforts and support in completing this work.

REFERENCES

[1] D. Limonadi "Smart lander reference surface scenario and reference vehicle description, rev 0", , Jet Propulsion Laboratory Interoffice Memorandum, November 2, 2001,

[2] T. Menzies, B. Cukic, H. Singh, and J. Powell, "Testing nondeterminate systems", *Proc. 11th Int'l Symp. Software Reliability Eng.*, San Jose, Calif. 2000, pp. 222-231.

[3] "NASA Mars exploration program, Mars 2007 smart lander mission, science definition team report", October 11, 2001

[4] D. R. Jeske and M. A. Qureshi, "Estimating the failure rate of evolving software systems", *Proc. 11th Int'l Symp. Software Reliability Eng.*, San Jose, Calif. 2000, pp. 52-61.

[5] E. T. Baumgartner, H. Aghazarian, A. Trebi-Ollennu, T. L. Huntsberger, and M. S. Garrett, "State estimation and vehicle localization for the fido rover," in *Proc. SPIE Conf. Sensor Fusion and Decentralized Control in Autonomous Robotic Systems III*, vol. 4196, Nov. 2000, pp. 329—336.

[6] B. Boehm and H. In, "Identifying quality-requirement conflicts," *IEEE Software*, March 1996, pp. 25-36.

[7] M. R. Barbacci, M. H. Klein, and C. B. Weinstock, "Principles for evaluating the quality attributes of a software architecture", Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University, March 1997.

[8] R. H. Thayer, "Software system engineering: a tutorial", *Computer*, April 2002, pp. 68-73.

[9] R. W. Butler and G. B. Finelli, "The infeasibility of quantifying the reliability of life-critical real-time software", *IEEE Trans. Software Eng.*, Vol. 19, No 1, Jan. 1993, pp. 3-12.

[10] D. Zeitler, "Realistic assumptions for software reliability", *Proc. Int'l Symp. Software Reliability Eng.*, IEEE CS Press, Loa Alamitos, Calif., 1991, pp. 67-74.