Entropy-Based Measure of Spacecraft Planned Activity Complexity: Applied to Mars Exploration Rovers Surface Activities

Ashitey Trebi-Ollennu
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
ashitey@jpl.nasa.gov

Abstract - *This paper describes a new methodology for the evaluation of the complexity of planned spacecraft activities by Earth-based operators. The methodology is based on a novel computation of the combined activity sequences entropy. The goal of this research is to develop a methodology which measures the degree of complexity of a spacecraft planned activity. For each activity command sequence, a sequence entropy is computed based on the concept of entropy from information theory. The overall planned activity complexity evaluation is computed using the Combined Activity Sequences Entropy, activity constraints, and the resources expended by the spacecraft planning team to build the command sequences. Finally, results from applying planned activity complexity evaluation to the Mars Exploration Rover mission robotic arm in-situ and mobility activities over a period of a 1000 sols are presented.*

Keywords: Mars Exploration Rover, planetary rovers, command sequencing, activity plan, spacecraft operations.

1 Introduction

The Mars Exploration Rover (MER) mission's Spirit and Opportunity rovers have successfully operated for over five Earth years, well beyond the originally designed surface lifetime of 90 sols (Martian days). The Mars rovers have grabbed hold of the public imagination with unprecedented mobile surface exploration on Mars, stunning close-up images of the Martian surface, and groundbreaking geological evidence of water-drenched environments in Mars's past.

The MER surface operations have evolved from the prime mission of 18 hours high-intensity Mars-Time operations (operations team and operators live on Mars-time) to 8 hours Earth-time operations (i.e., getting operators off Mars-time schedule) over 1200 sols [1]. The impetus for this evolution was the realization that the rovers could survive several orders of magnitude beyond their slated 90 sols design lifetime. This dictated the need for the development of a surface operations model that considers Human Factors and is sustainable indefinitely within the project's resource constraints.

The approach adopted included progressive automation of the ground processes to the extent possible thus reducing the operations team workload. In addition, the complexity of a sol's plan was severely curtailed by eliminating some parallel spacecraft activities that may demand the operations team to conduct resource intensive progressive elaboration planning to prevent potential onboard resource conflicts [1]. The collective implementation of the above strategies resulted in fewer hours needed for the tactical planning process for a sol. However, there have not been any efforts to objectively measure and confirm a corresponding reduction in the complexity of a sol's plan due to the lack of an objective methodology for computing activity plan complexity. This research addresses this technology gap by developing an objective quantitative metric to compute a measure of command sequence complexity and a spacecraft activity plan that will enable absolute comparison between different command sequences independent of spacecraft and sequence language and language format.

The sections of this paper are organized as follows: Section 2 presents a brief description of the command sequence entropy formulation. Section 3 presents a description of the Planned Activity Complexity Evaluation formulation. Section 4 presents results of the Planned Activity Complexity Evaluation applied to the robotic arm in-situ and mobility activities for the Spirit and Opportunity rovers during the surface operations phase of the MER mission. The paper closes with conclusions in Section 5.

2 Command sequence entropy formulation

The rover command sequence generation can be considered to be similar to software code development, where each command and its specified arguments represent a line of software code [2]. Each command sequence represents several hundred lines of code. Management and engineers frequently have to measure the degree of software structural complexity, however the large size of modern software systems makes manual evaluation impractical, and subjective evaluations are vulnerable to bias. In the literature, software complexity has been formulated as the degree of difficulty and resources needed in analyzing, maintaining, testing, designing and modifying the software [3-7]. The IEEE Standard Computer Dictionary [8] defines complexity as: "(Apparent) the degree to which a system or component has a design or implementation that is difficult to understand and verify."

In this research we will borrow from the software engineering literature the definition of complexity. The complexity of a command sequence will be considered as a broad measure of the following:

1. *The complexity of the Activity Plan, which is the inherent complexity, created during Activity Planning.*
2. *The resources needed to translate the Activity Plan to command sequences, the resources have at least two aspects: time (i.e., man-hours to build and verify the command sequence) and inherent degree of complication (i.e., intricacy of conditional branches, degree of nesting of command sequences, etc.).*

Combined Activity Sequences Entropy (CASE) addresses the complexity measure for the inherent degree of complication of a command sequence. The Planned Activity Complexity Evaluation (PACE) measure addresses the resources required (e.g., man-hours, etc) to build and verify the command sequence [9].

The input to the Sequence Entropy model is a Command Usage Effort (CUE) for each command in the command sequence. A CUE (λ) represents the amount of work measured in information units the operator has to input to use the particular spacecraft command. CUE effectively captures the analytical work that is required to select associated arguments and control flags of a particular spacecraft command. CUE information unit has a range of rating levels from, "LOW" to "VERY HIGH." The CUE rating level expresses the weighted impact a particular spacecraft command usage has on the command sequence development resources. Each rating has a corresponding real value weight derived from the degree to which the factor can influence command sequence generation resources. A CUE rating of LOW denotes a command that provides sufficient information about itself such that very little or no CUE information unit is required from the user. In other words very little or no analytical work is required to select associated arguments and control flags for that particular spacecraft command. A CUE rating of VERY HIGH denotes a command that has sparse self-information and therefore requires the user to provide several CUE information units. A rating of VERY HIGH indicates that the analytical work required in selecting associated arguments and controlling flags for the command is inherently complex and resource intensive.

The CASE ($H(S)$) is defined as follows. Let $S = \{s_i, i = 0, \ldots, n\}$ be a command sequence with a backbone sequence s_0 and zero or more helper sequences s_i. Each sequence s_i, is an ordered list of commands $c_j \in C$. A λ_i (CUE) is associated to a command c_j via a command dictionary C. To compute the command sequence entropy the relative frequency of occurrence of λ_i is computed first, taking into account the lexical scope where λ_i appears in the command sequence. This is done so that the CASE (backbone and helper sequences) is not skewed.

Once an appropriate measure of the relative frequency of each λ_i is found, we compute its relative weight as a function of its frequency: $w_i = \phi(\lambda_i)*\rho(\lambda_i)$, where ϕ is a function that maps λ_i information units to its corresponding frequency and ρ is the rating model used.

Using the relative weight w_i, the CASE is defined as follows:

$$H(S) = \sum_i \hat{w}_i \log\left(\frac{1}{\hat{w}_i}\right) \quad H(S) \in [0,1] \qquad (1)$$

where

$$\hat{w}_i = w_i \Big/ \sum_i w_i \quad \hat{w}_i \in (0,1]. \qquad (1a)$$

3 Planned activity complexity evaluation (PACE) formulation

The overall PACE is computed using as input CASE, activity constraints and the resources (time) expended by the spacecraft planner to build the command sequences [11]. The PACE measure addresses the resources required (e.g., man-hours etc) to build and verify the command sequence. These are resources needed to translate the *Activity Plan* to command sequences, the resources have at least two aspects: time (i.e., man-hours to build and verify the command sequence) and inherent degree of complication (i.e., intricacy of conditional branches, degree of nesting of command sequences, etc.). Command Sequence entropy formulation captures one axis of the resource complexity which is the inherent degree of complication of a command sequence. The other resource complexity axis is captured by the PACE measure which incorporates the resources required (e.g., man-hours, etc) to build and verify the command sequence.

Resource Impact Drivers that have a multiplicative effect [10] on resource utilization in the completion of the command sequences development process must be identified. The selection of multiplier factors is project specific but must be based on a strong rationale that can independently explain its significance as a multiplicative effect on resource utilization during the command sequence development process. The PACE (Ω) for an *Activity Plan* is defined as:

$$\Omega = U \cdot H(S) \quad \Omega \in [0,1] \tag{2}$$

where

$$U = \sum_{j=1}^{m} \mu_j \hat{R}_j \quad U \in [0,1],\ \hat{R}_j \in [0,1],\ \sum_j \mu_j = 1 \tag{2a}$$

and $j = 1, \ldots, m$ is the number of Resource Impact Drivers identified. $\hat{R}_j$ is a unitless measure of the jth Resource Impact Driver, because of the plurality of measurement scales for Resource Impact Drivers, it is necessary to normalize Resource Impact Drivers into a similar range with unitless measures. μ_j is the subjective relative weight that reflects the degree of influence of jth Resource Impact Driver and U represents the weighted sum of all Resource Impact Drivers that have a multiplicative effect on resource utilization in the completion of the command sequences development process.

4 PACE applied to MER robotic arm and mobility activities

4.1 MER instrument deployment device (IDD) and mobility CASE computation

The IDD and mobility commands CUE information unit were rated using a linear function (see Table 1) from "LOW", "LOW+25", "LOW+50", "LOW+75", "HIGH", "HIGH+25", "HIGH+50", "HIGH+75" and "VERY HIGH". Over ninety percent of the VERY HIGH CUE rated IDD and mobility commands are used to set parameters and are seldom used in regular command sequencing. The rating model $\rho: \lambda \rightarrow \Re$ is a monotonically decreasing function (see Table 1).

Table 1. Rating Model.

CUE	$\rho(\lambda_i)$
LOW	0.975
LOW+25	0.850
LOW+50	0.725
LOW+75	0.600
HIGH	0.500
HIGH+25	0.375
HIGH+50	0.250
HIGH+75	0.125
VERY HIGH	0.025

Figures 1 to 4 show the MER IDD CASE results. The IDD CASE trend for both Spirit and Opportunity demonstrate that the inherent degree of complexity of IDD activities have remained the same throughout the surface mission regardless of changing project resources. There has always been anecdotal evidence to support this observation. However, CASE results present the first empirical documented evidence of the incredible way the MER surface operations has maintained a constant activity load within tight project resource constraints. The CASE results contradicts the established notion that one of the key factors in reducing MER surface operations to 8 hours Earth-time was the curtailing of the complexity of a sol's plan.

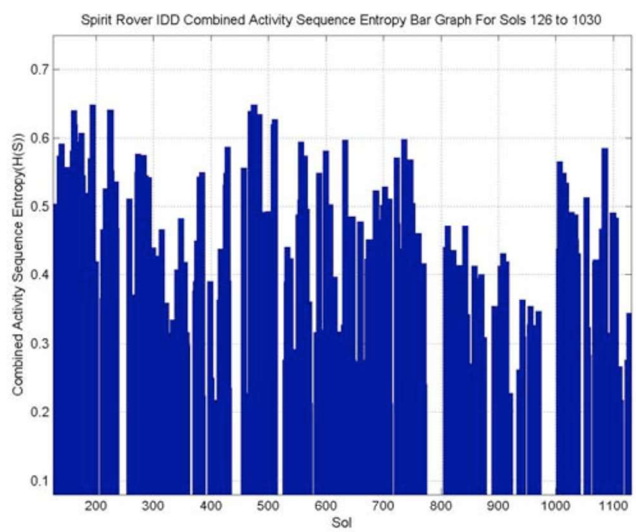

Figure 1. Spirit's IDD CASE from sols 126 to 1030.

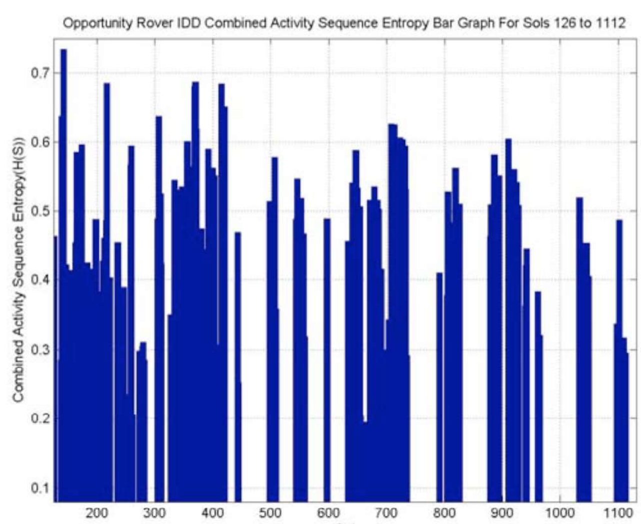

Figure 2. Opportunity's IDD CASE from sols 126 to 1112.

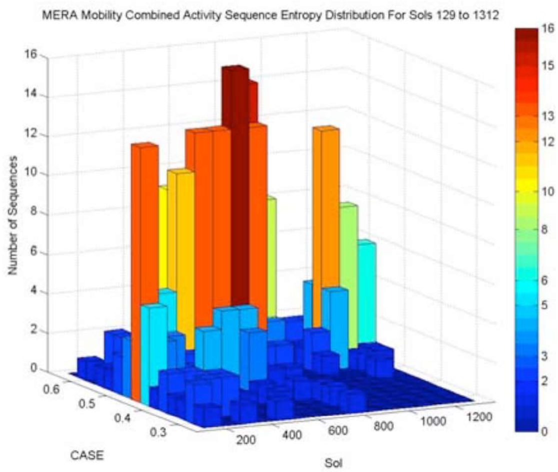

Figure 3. Spirit's mobility CASE distribution from sols 126 to 1200.

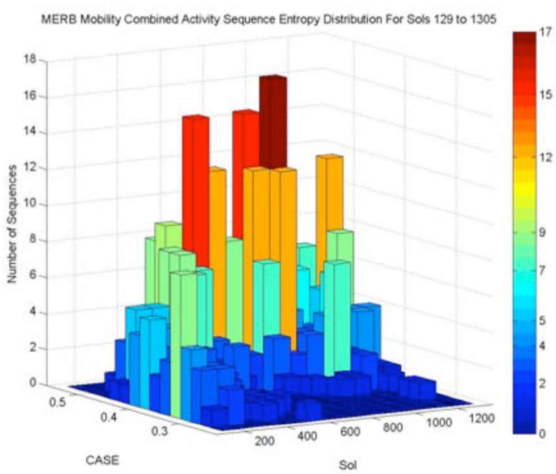

Figure 4. Opportunity's mobility CASE distribution from sols 126 to 1200.

Figure 1 shows that between sols 180 and 514 is primarily where the maximal CASE for Spirit is located. This is not surprising since it was during this period that Spirit arrived at a treasure trove of bedrock and rocks altered by the presence of liquid water (the sites are The West Spur and Cumberland Ridge). These rocks had strikingly different morphology from the basaltic rocks that Spirit had seen on the plains. As a result extensive in-situ investigations were performed on these rocks with the Athena payload [9] on the IDD. During this period Spirit performed more complex IDD operation (i.e., command sequences) against increasingly challenging terrain (e.g., slopes, rock roughness, etc.).

In Figure 2, a similar graph is displayed for Opportunity with the maximal CASE located between sols 132 to 315. This period corresponds to when Opportunity was in Endurance Crater, a 160m wide crater. In Endurance crater an intense in-situ study of bedrock exposed by the crater formation was conducted. In Endurance Opportunity performed its most complex IDD work at slopes greater that 25degrees, a first for both rovers. Comparing Spirit and Opportunity rovers IDD CASE distribution (Figures 1 and 2), Opportunity shows a drastic decline after sol 315 that can be attributed to the fact that Opportunity was back on the plains after its successful egress from Endurance crater and is en route to Victoria crater. This also reflected in an increase in mobility CASE (see Figure 4) for Opportunity as will be expected.

4.2 MER IDD and mobility PACE computation

Imposed schedule (duration or time) and the operations team experience (i.e., Plan Activity Experience Weighting) were considered to be the only resource impact drivers that had a multiplying effect on resource utilization for IDD and mobility command sequence generation.

PACE for IDD and mobility command sequence generation was defined as follows:

$$\Omega_k = H(S_k) \times \kappa_k \tag{3}$$

On the MER mission, a schedule is imposed on the command sequence generation process, in this paper the imposed schedule is expressed as a duration range, where Max_Dur and Min_Dur represents the maximum and minimum allowable duration for the command sequence(s) generation for a particular sol's activities respectively. If the actual duration for developing the command sequence with CASE $H(S_k)$ is greater than the minimum duration allowed then κ_k is define as follows:

$$\kappa_k = \left(1 + \left[(1 + f_k) \times \left[\frac{Max\_Dur - Actual\_Dur_k}{Actual\_Dur_k - Min\_Dur}\right]\right]\right) \tag{4}$$

where k is the sol number and $f \in (0, 1]$ is the plan activity experience weighting; otherwise, κ_k is define as:

$$\kappa_k = \left(1 + \left[(1 + f_k) \times \left[\frac{Min\_Dur - Actual\_Dur_k}{Max\_Dur - Actual\_Dur_k}\right]\right]\right) \tag{5}$$

In order to avoid singularities in the computation of Ω_k, if the actual duration for CASE $H(S_k)$ is within a pre-defined threshold above the Min_Dur, the duration multiplier in the κ_k expression is set to zero.

Figures 5 and 6 depict Spirit and Opportunity rovers IDD and mobility PACE over 1600 sols. Since the CASE for each rover has remained fairly constant and the duration assigned to complete the command sequence was reduced, it was expected that the PACE would be trending upward over the lifetime of MER surface operations. Figures 5 to 8 confirm this trend and show that IDD and mobility Activity Plans for both rovers have been steadily increasing in complexity as a result of limited resources, e.g., shorter duration to generate IDD and mobility commands and multiple-sol command sequences. The trend shows almost a doubling of the complexity of IDD and mobility activity plans from sol 300 onwards for both rovers.

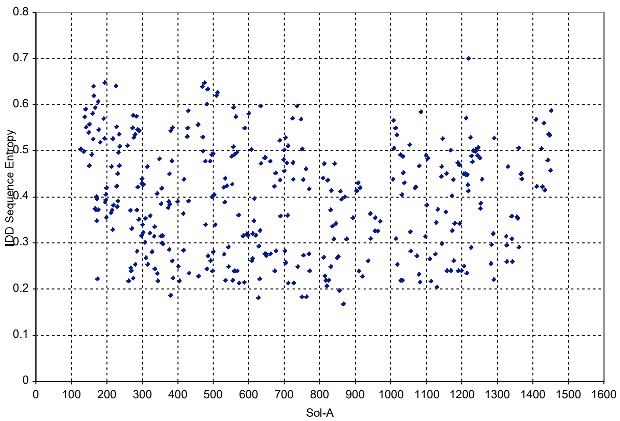

Figure 5. Spirit's IDD PACE from sols 126 to 1600.

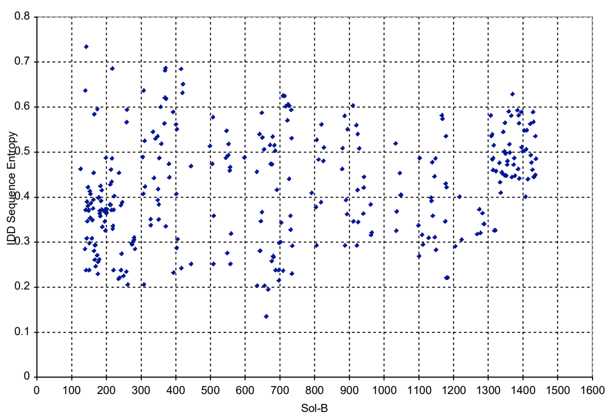

Figure 6. Opportunity's IDD PACE from sols 126 to 1600.

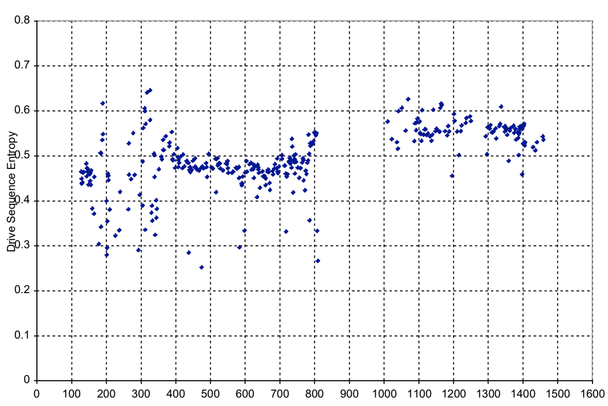

Figure 7. Spirit's mobility PACE from sols 126 to 1600.

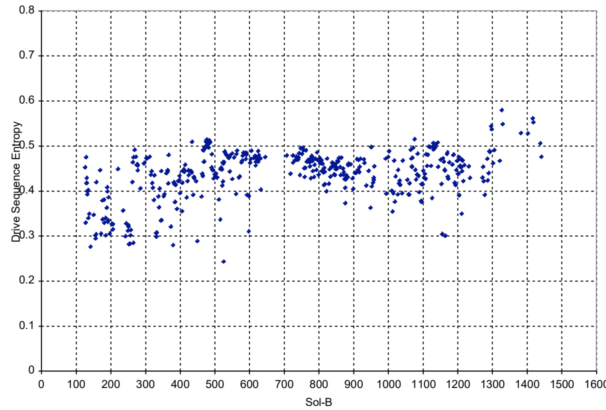

Figure 8. Opportunity's mobility PACE from sols 126 to 1600.

Can the increase in IDD activity complexity be explained by the experience curve, which states that the more often a task is performed the lower the resource cost of doing it? The learning curve by contrast states that the more times a task is performed, the less time is required on each subsequent iteration. So in effect the learning curve effect and the experience curve effect express the relationship between experience and efficiency.

We did take into account the experience of the team, and as a result only 70% of the resource usage for the IDD and mobility command sequence generation from sol 300 onwards was used in the computation of PACE for each of the rovers. This is quite extraordinary since the experience curve is generally believed to account for about 20% reduction in resource cost each time the cumulative output of the task doubles. The cumulative output of in-situ activity using the IDD and mobility on Mars is science return, a quantity not easily measured because it is subjective. However, it is generally understood that the MER rovers' science return is not directly proportional to the number of planned IDD and mobility in-situ activities. As a result, the doubling of PACE for the sol period under investigation cannot be solely attributed to the experience curve effect.

However, the learning curve effect could be the potential explanation for obtaining this incredible efficiency in doubling the complexity of IDD activity plans for both rovers from sol 300 onwards, not accounting for the fact that the rovers are aging and any hardware failures may change the mission profile, impact the surface operations process and, as a result, may reset the learning curve.

This however, raises an important question, can the current PACE of IDD level of activities be sustained without negatively impacting the project team morale or result in team burn out?

5 Conclusions

PACE can be used, either to evaluate the complexity of a Planned Activity, or as a tool to monitor the workload of spacecraft operators and identify any developing trends in the spacecraft operations. PACE and CASE can also be used to identify key autonomous technologies for spacecraft operations that will significantly lower the workload of Earth-based operators thereby reducing operations cost and increasing science return. The expectation is that PACE metrics can provide useful feedback to mission planners to enable them make informed decisions based on historical data during mission concept development, mission design, and mission operations architecture development. Without this feedback, many decisions will be made ad hoc.

Acknowledgment

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. I owe a special debt of gratitude to Dr. Antonio Diaz-Calderon for his help and contribution to the development of the algorithms applied to the MER IDD activities in 2007 and the resulting JPL publication.

References

[1] A.H. Mishkin, D. Limonadi, S.L. Laubach and D.S. Bass, "Working the Martian night shift - the MER surface operations process," *IEEE Robotics & Automation Magazine*, Vol. 13, Issue 2, pp. 46-53, 2006.

[2] S. Maxwell, B. Cooper, F. Hartman, J. Wright, J. Yen and C. Leger, "The best of both worlds: Integrating textual and visual command interfaces for Mars rover operations," Proc. IEEE International Conference on Systems, Man, Cybernetics, Waikoloa, HI, vol. 2, pp. 1384–1388, 2005.

[3] R. McEliece, *The theory of information and coding*, Cambridge University Press, New York, NY, 1984.

[4] S.H. Kan, *Metrics and models in software quality engineering*, Addison-Wesley Publishing Company, Reading, MA, 1995.

[5] W. Harrison, "An entropy-based measure of software complexity," *IEEE Trans. on Software Engineering*, Vol. 18, No. 11, pp. 1025-1029, Nov. 1992.

[6] R.K. Lind and K. Vairavan, "An experimental investigation of software metrics and their relationship to software development effort," *IEEE Trans. on Software Engineering*, Vol. 15, No. 5, pp. 649-653, May 1988.

[7] R.D. Banker, S.M. Datar, C.F. Kemerer and D. Zweig, "Software complexity and maintenance costs," *Communications of the ACM*, Vol. 36, No. 11, pp. 81-95, Nov. 1993.

[8] Institute of Electrical and Electronics Engineers, *IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries*, New York, NY, 1990.

[9] A. Trebi-Ollennu and A. Diaz-Calderon, Planned Activity Complexity Evaluation (PACE): Applied to Mars Exploration Rovers surface activities, JPL Technical Report, JPL Publication 07-4, July 2007.