JPL Publication 07-4

Planned Activity Complexity Evaluation (PACE): Applied to Mars Exploration Rovers Surface Activities

Ashitey Trebi-Ollennu and Antonio Diaz-Calderon
Mobility and Manipulation Group, Mobility and Robotic Systems Section (347)
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

11 July 2007

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration We also owe a special debt of gratitude to Sharon L. Laubach, MER sequencing team chief.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

An Activity Plan for a spacecraft consists of a time-ordered set of engineering and science activities to be performed by the spacecraft over a specified time period (hours, days, weeks, months etc). Methodologies for measuring the degree of complexity of spacecraft Planned Activities by Earth-based operators is lacking in the spacecraft operations literature. 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 (CASE). A command sequence (or sequence) is an ordered list of commands with associated arguments and control flags that will be executed by the spacecraft onboard sequence engine. Each activity in the Activity Plan is expanded into command sequence, and may comprise multiple command sequences. 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 (SE) is computed based on the concept of entropy from information theory. The overall Planned Activity Complexity Evaluation (PACE) is computed using the Combined Activity Sequences Entropy (CASE), activity constraints and the resources (e.g. time) expended by the spacecraft planning team to build the command sequences. Finally, results from applying PACE to the Mars Exploration Rover (MER) mission robotic arm in-situ activities over a period of 1000 sols are presented.

ACKNOWLEDGMENTS

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We also owe a special debt of gratitude to Sharon L. Laubach, MER sequencing team chief.

TABLE OF CONTENTS

1 Introduction

In January 2004, NASA landed the twin rovers Spirit and Opportunity on opposite sides of Mars, initiating the Mars Exploration Rover (MER) mission. 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. By 2 January 2006 and 24 January 2006, respectively, Spirit and Opportunity had successfully operated for two Earth years on the surface of Mars, well beyond the originally designed surface lifetime of 90 Martian days. In addition, they have sent back over 140,000 images and 23 GB of data and explored over 6.5 km of Martian terrain per rover. MER is by far the most successful and publicly engaging Mars mission ever flown. MER is also one of the most complex planetary robotics systems ever conceived, developed, and deployed in the history of planetary exploration.

The MER surface operations has evolved from the prime mission of 18 hours high-intensity Mars-Time (operations team and operators live on Mars-time) operations to 8 hours Earth-time (i.e. getting operators off Mars-time schedule) operations 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 process of developing a command sequences to send to an unmanned planetary spacecraft has always been time-consuming and labor-intensive. 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 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 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. The noted mathematician, Lord Kelvin once said: "If you can measure something and put a number to it, you can begin to understand it. If you cannot measure it, you have a very sorry ability to understand it." 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, sequence language and language format.

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 **C**ombined **A**ctivity **S**equences **E**ntropy (**CASE**). A command sequence (or sequence) is an ordered list of commands with associated arguments and control flags that will be executed by the spacecraft onboard sequence engine. An Activity Plan for a spacecraft consists of a time-ordered set of engineering and science activities to be performed by the spacecraft over a specified time period (days, weeks, months, etc). Each activity in the Activity Plan is expanded into command sequence, and may comprise multiple command sequences. 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 **S**equence **E**ntropy (**SE**) is computed based on the concept of entropy from information theory [2]. The overall **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) is computed using the **C**ombined **A**ctivity **S**equences **E**ntropy (**CASE**), activity constraints and the resources (time) expended by the spacecraft planner to build the command sequences.

The sections of this paper that follow are organized as follows: Section 2 presents a brief description of the command sequence entropy formulation. Section 3 presents a detailed description of the **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) formulation. Section 4 presents results of **PACE** applied to the robotic arm in-situ activities for 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 [3]. Therefore 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 [4,5,6,7]. 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." Evans et. al. [9] also define complexity: "(Inherent) the degree of complication of a system or system component, determined by such factors as the number and intricacy of interfaces, the number and intricacy of conditional branches, the degree of nesting, and the types of data structures." Software complexity measures have been found in general to provide a more accurate measure of a software program's structural complexity than counting lines of code. Software complexity measurements have also been found to facilitate comparison between different algorithms or designs and also provide indirect estimation and prediction for the number of inherent and remaining bugs and the staff resources required for software development. Software complexity measurements can also be used as a direct measure of the software project progress and quality during the life cycle of the software project [10].

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 **A**ctivity **S**equences **E**ntropy (**CASE**) addresses the complexity measure for the inherent degree of complication of a command sequence. The **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) measure addresses the resources required (e.g. man-hours, etc) to build and verify the command sequence.

The input to the Sequence Entropy model is a **C**ommand **U**sage **E**ffort (**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. If a linear rating model is used for **CUE** information unit, a LOW rating will correspond to 0 and VERY HIGH rating will correspond to 1. The rating scale however can be any linear or nonlinear function so long as it is based on a strong rationale that can independently explain the significant impact on command sequence development resources for each spacecraft command.

An Activity Plan command sequence may consist of several nested sequences with a main command sequence termed the "backbone sequence" calling several other command sequences termed "helper command sequence". Table 1 shows an example of an Instrument Deployment Device (IDD) command sequence

Table 1. An example of IDD nested command sequence, dx0x1 is the backbone and nxxx1, p1xxx and p4xxx are helper command sequences.

Command Number	Command Sequence ID	Spacecraft Commands	Comments
1	dx0x1	Unstow IDD	
2	dx0x1	IDD Move Place Tool on Rock	
3	dx0x1	RUN Command Sequence nxxx1	RUN MB Command Sequence to collect Spectra
4	dx0x1	Move IDD from rock	
5	dx0x1	RUN Command Sequence p1xxx	RUN Camera Command Sequence to acquire image of rock in IDD work volume
6	dx0x1	IDD Move Place Tool on Soil	
7	dx0x1	RUN Command Sequence p4xxx	Run Microscopic Image command sequence to acquire images of soil
8	dx0x1	IDD Move To CCT	
8	dx0x1	Stow IDD	

The **C**ombined **A**ctivity **S**equences **E**ntropy (**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 **C**ombined **A**ctivity **S**equences **E**ntropy (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 **C**ombined **A**ctivity **S**equences **E**ntropy (**CASE**) is defined as follows:

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

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

3 Planned Activity Complexity Evaluation (PACE) Formulation

The overall **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) is computed using as input **C**ombined **A**ctivity **S**equences **E**ntropy (**CASE**), activity constraints and the resources (time) expended by the spacecraft planner to build the command sequences. The **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) measure addresses the resources required (e.g. man-hours etc) to build and verify the command sequence. The resources needed to translate the **A**ctivity **P**lan 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 **P**lanned **A**ctivity **C**omplexity **E**valuation (**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 it's significant as a multiplicative effect on resource utilization during command sequence development process. An example of a list of multiplier factors is as follows (this is not an exhaustive list.);

1. **Schedule Weighting**

This is a measure of the degree to which the schedule imposed by the project to develop the command sequences for the activity plan was satisfied.

2. **Activity Plan Execution Resource Constraint Weighting**

This is a measure of the degree to which the developed command sequence(s) satisfies the imposed execution resources (time, data volume, etc) assigned to the activity plan.

3. **Tool Experience Weighting**

This measures the level of experience of the spacecraft operations team developing the command sequence with the sequencing tools or relative unfamiliarity with the sequencing tools.

4. **Plan Activity Experience Weighting**

This measures the operations team experience developing command sequences for a particular activity plan (highly capable analysts). Weighting is defined in terms of team experience building command sequences for this type of activity.

The **PACE** (Ω) for an **A**ctivity **P**lan is defined as:

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

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

$j = 1,\ldots,m$ 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_i 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 *In Situ* Surface Activities

Daily tactical uplink surface operations for MER starts with the science and engineering teams using the Science Activity Planner (SAP) [11] to collaboratively create an Activity Plan during the Science Operations Working Group meeting. The Activity Plan consists of a time-ordered set of engineering and science activities to be performed by the rover over a specified time period, usually a Sol (a Martian day, which corresponds to 24 hours, 39 minutes). When the Mission Manager approves the Activity Plan it is subsequently called a Validated Activity Plan. The next step in the process is to partition the Validated Activity Plan and assign specific activities to members of the Integrated Sequencing Team (IST) to expand into command sequences. The IST processes are designed to avoid the significant impact of "Parkinson's Law;" i.e., work expanding to fill (and often exceed) the time allowed. The Instrument Deployment Device (IDD) and Mobility activities from the Validated Activity Plan are assigned to the Rover Planners (RP) or Rover Drivers. RP must reconcile two conflicting aspects of the process -- the increasingly important need for speed in the command sequence generation and delivery and the equally important need for safety of the rovers. The Rover Planners generate the command sequence for Mobility and IDD activities that will be executed by the rover onboard sequence engine. The command sequences built by the RP may “activate" or "run" other command sequences (e.g. cameras, etc.) which may run simultaneously with the RP command sequence. The RP coordinates with the rest of the IST team to determine the number of sequence(s) required for an activity or activities to satisfy science priorities (e.g., time of day to make observation, etc), avoid conflicts (e.g., resource usage), comply with flight rules, and/or manage resource constraints.

Like many other operations processes, the MER command sequence generation process has its constraints – and bottleneck(s) and is the driving factor in how the resources are managed. The RP command sequence generation for the IDD and Mobility are more often than not the bottleneck of the MER surface operations Integrated Sequence Team command sequence generation process [12]. We posit that a measure of the **P**lanned **A**ctivity **C**omplexity **E**valuation (**PACE**) of IDD activities over the operational life of the MER rovers Spirit and Opportunity will be an excellent reflection of the MER rovers' surface Activity Plan complexity.

4.1 IDD Command Sequence Generation

Spirit and Opportunity are equipped with a 5 degree-of-freedom Instrument Deployment Device (IDD), a dexterous robotic manipulator (Figure 1). At the end-effector of the IDD is mounted a unique in-situ instrument suite called the Athena science payload [13] (Figure 2) that has been designed to measure and understand the detailed geochemistry and morphology of the surface of Mars. The in situ instrument suite includes a Mössbauer Spectrometer (MB), an Alpha Particle X-ray Spectrometer (APXS), a Microscopic Imager (MI), and a Rock Abrasion Tool (RAT). The IDD provides a dexterous in-situ manipulation capability to enable the placement and holding of the instruments directly against rock and soil targets of interest within the IDD work volume, thus allowing for detailed inspection of rocks and soil to reveal their elemental and mineralogical composition and document their geological time history of water.

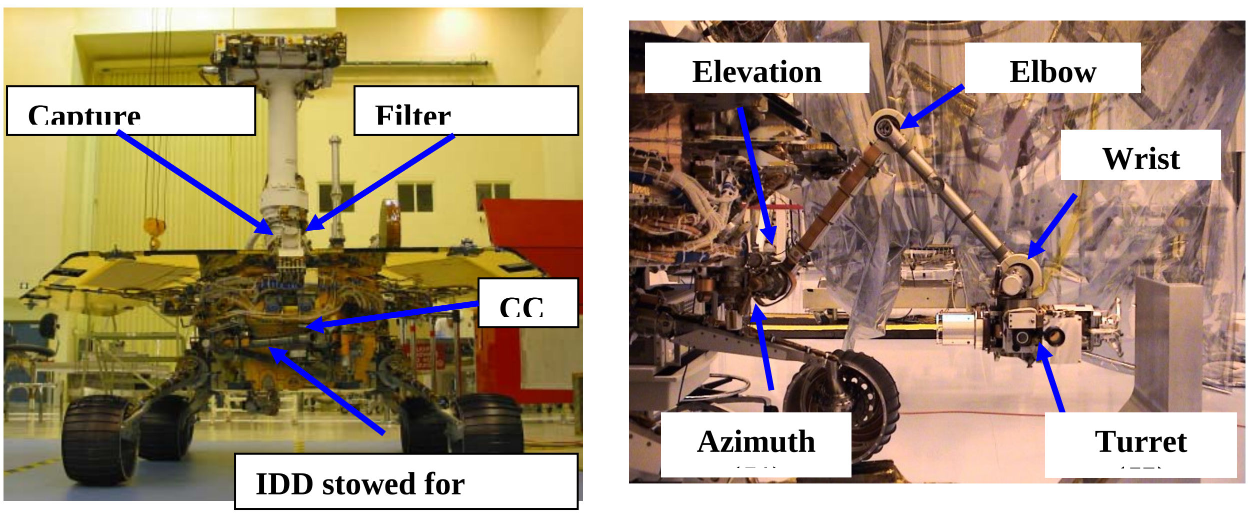

Figure 1. IDD stowed for driving and rover-mounted targets, left. Unsowed IDD with joints labeled, right.

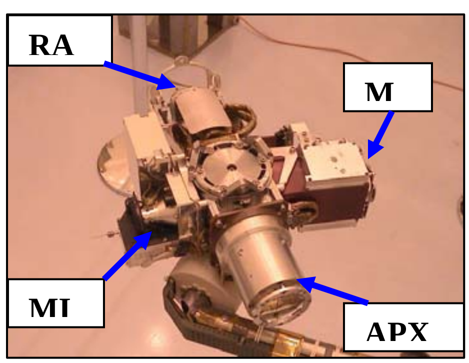

Figure 2. Layout of Athena science payload mounted on the IDD turret.

We present a brief overview of the IDD. For detailed information on the IDD mechanical system and software, the interested reader is referred to [13, 14]. The IDD is mounted at the front shelf of the belly of the rover's Warm Electronic Box as shown in Figure 1. The IDD weighs approximately 4kg and carries a 2 kg Athena Science payload. The IDD is stowed as shown in Figure 1 (left image) during rover traverses. Figure 1 (right) depicts the IDD mechanical system with all five joints labeled. Figure 2 depicts the Athena Science Payload as mounted on the IDD turret. Each of the in-situ instruments is equipped with a proximity sensor (contact sensor) to detect contact between the instrument and any target, be it the rover or Martian surface. For the MI, MB and RAT, the contact sensors are configured to be dual redundant per instrument. The APXS, however has two independent contact sensors. The first contact sensor is used to detect successful opening or closing of the APXS dust door and the second is only activated if the door is latched open as the contact plate is depressed farther. There are three rover-mounted targets as shown in Figure 1 (left image), namely, the Compositional Calibration Target (CCT) and the Capture and Filter Magnets. The Capture and Filter permanent magnets are designed to attract dust from the Martian atmosphere. The CCT is a magnetite calibration target for the MB. It is also used to open the APXS dust door for APXS placement on soil targets. A fourth target is mounted on the forearm of the IDD and it is called the RAT brush station.

IDD command sequence generation is the most complex task of all rover sequence planning. A typical IDD sequence has hundreds of commands and is highly integrated with other rover activities. There is virtually no automation of IDD command sequence generation beyond the use of sequence "Macros". A Macro is a prototype version of an ordered list of commands with associated arguments and control flags that perform specific functions (e.g., Unstow or Stow the IDD, etc.). Macros are used to capture functions that will be repeatedly used during IDD operations and, in general, require little or no modifications to their template, but may require changes to parameters and control flags associated with their commands. In the case of the IDD there is a unique set of Macros for each rover. All IDD teach points (rover- mounted targets) operations are captured in Macros, e.g., APXS doors opening at the CCT, MB placement on the Capture Magnet, etc.

IDD sequences are event-driven; that is, status of the execution control flag of commands determines the behavior of the sequence engine. In order words, the successful completion of preceding commands in a sequence triggers the execution of the next command, and this process continues until the last command in the sequence is executed. However, if any command in sequence terminates with failure, the sequence is halted by the sequence engine. Another way to think of event-driven sequences is that commands do not have time tags associated with them that indicate when they are to be dispatched.
IDD sequences also employ conditional sequencing with logical conditionality. This enables the RP to sequence the IDD such that a secondary set of commands (e.g., preclude IDD activities) would be executed depending on whether predefined conditions are not satisfied. For MER the variables that are compared in conditional sequencing constructs are called Defined Data Items (DDI). IDD conditional sequences also employ a generic DDI called LAST_COMMAND_STATUS (available to all subsystems) with values SUCCESS/FAILURE.

4.2 MER IDD CASE Computation

The IDD commands **CUE** information unit were rated using a linear function (see Table 2) 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 commands are used to set parameters and are seldom used in regular command sequencing.

Table 2. CUE Rating Distribution of IDD Commands

CUE RATING (λ_i)	Number of IDD Commands
LOW	41
LOW+25	11
LOW+50	9
LOW+75	5
HIGH	16
HIGH+25	2
HIGH+50	3
HIGH+75	2
VERY HIGH	56

The rating model $\rho : \lambda \rightarrow \Re$ is a monotonically decreasing function (see Table 3).

Table 3. 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 3 and 4 depict Bar Graphs of Spirit and Opportunity rovers IDD Combined Activity Sequence Entropy (**CASE**) for Sols 126 to 1112. As expected there are peaks and valleys in the **CASE** profile for this period, the valleys represent the periods of drive (traverse) campaigns - i.e., periods when the exploration emphasis was on driving to a particular geological site. Once the rover arrives at the geological site the exploration emphasis is focused on intensive In-situ investigation using the IDD resulting in the peaks. The results from Figures 3 and 4 also agreed with the subjective impression of the MER project management team that there has not been any significant reduction in complexity of a sol's plan as result of moving to Earth-time planning. It can be inferred from the distribution that the inherent degree of complexity of IDD activities for the Spirit rover has remained the same (uniform distribution) throughout sols 126 to 1030.

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.

Figures 5 and 6 depict a 3D histogram of Spirit and Opportunity rovers' CASE distribution respectively. From Figures 5 and 6 the two variables CASE and Sol define the location in the plane, the number of sequences the height of corresponding column, the fourth its color (the more intense the color, the higher the CASE).

Figure 5 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 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 6, 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 5 and 6), 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. At Meridiani Planum, Opportunity's landing site, there were very few chances to perform opportunistic science since the terrain is featureless and strewn with several large fissures, and small and larger craters several hundreds of meters apart. In contrast at Gusev Crater, Spirit's landing site, there is a rock-strewn plain with low hills suitable for periodic opportunistic in-situ science - i.e., performing quick in-situ analyses called "touch-and-go" operations during traverse. These opportunistic science activities in general comprise the same set of in situ activities. This regular opportunistic science activity accounts for Spirit's fairly uniform distribution of CASE.

Table 4 shows the summary statistics for Spirit and Opportunity's CASE. As would be expected, their statistics are similar except for Opportunity having the most complex command sequence for the period analyzed.

Table 4. Summary Statistics of IDD CASE for Spirit and Opportunity Rovers

	Summary Statistics of Spirit and Opportunity Rovers IDD CASE				
Rover	Mean	Standard Deviation	Mode	Max	Min
Spirit	0.392	0.123	0.240	0.648	0.168
Opportunity	0.407	0.124	0.238	0.734	0.135

One factor not taken into account is that the rovers are aging and any hardware failures may change the mission profile, and might impact the surface operations process. Spirit and Opportunity have experienced hardware failures: Spirit's right front drive wheel actuator failure and Opportunity's a right front steering actuator failure, and IDD azimuth joint actuator significant degradation. However, these failures have not impaired the science return from both rovers and have had very small impact on IDD command sequence complexity; but future hardware failures might pose significant challenges to command sequence complexity.

The rover flight software has been upgraded several times from R9.0 to R9.1 and the last R9.2. The R9.1 upgrade mainly fixed several bugs in the flight software and improved mobility performance. However, R9.2 added new mobility and IDD capabilities. The new capabilities have not yet reduced IDD command sequence generation.

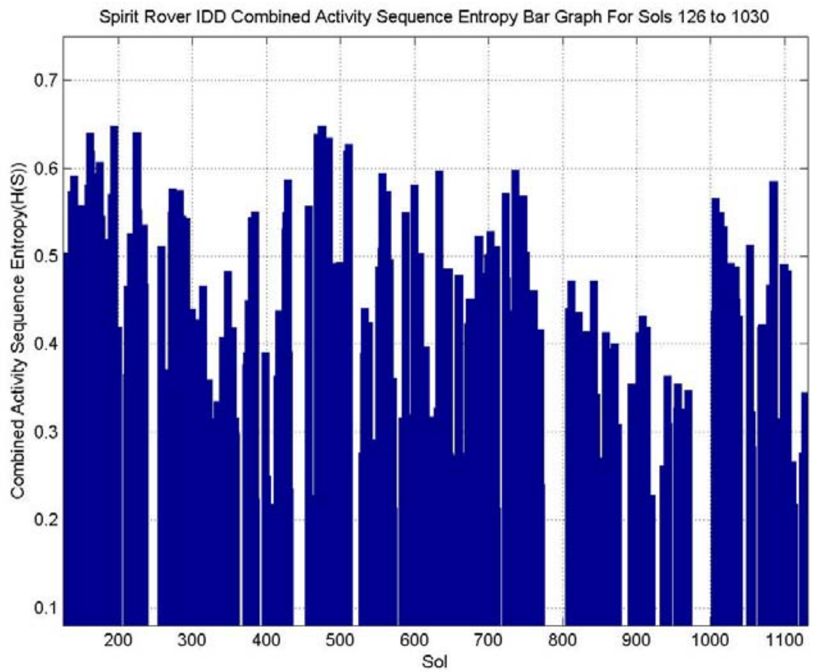

Figure 3. Spirit's IDD Combined Activity Sequence Entropy from Sol 126 to 1030

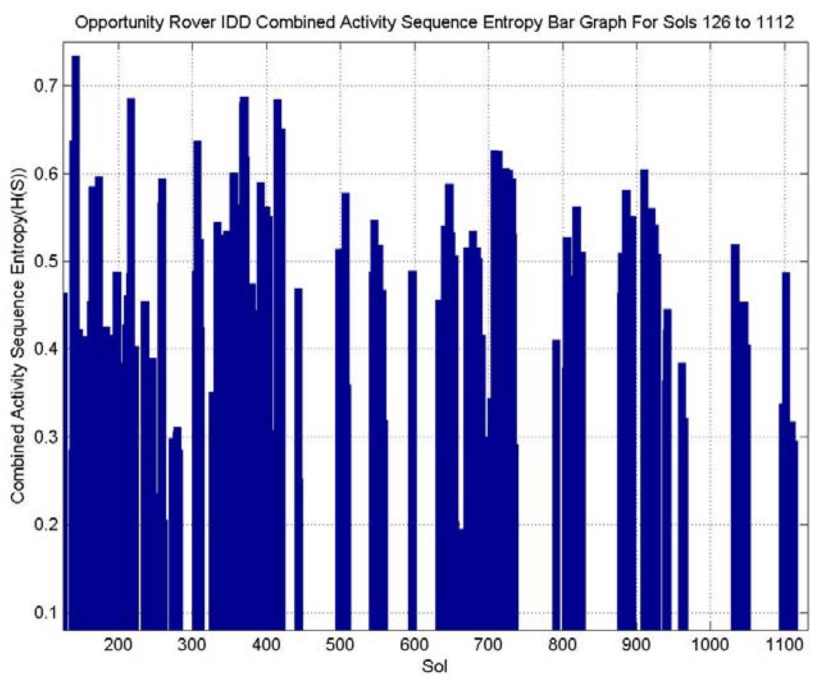

Figure 4. Opportunity's IDD Combined Activity Sequence Entropy from Sol 126 to 1112

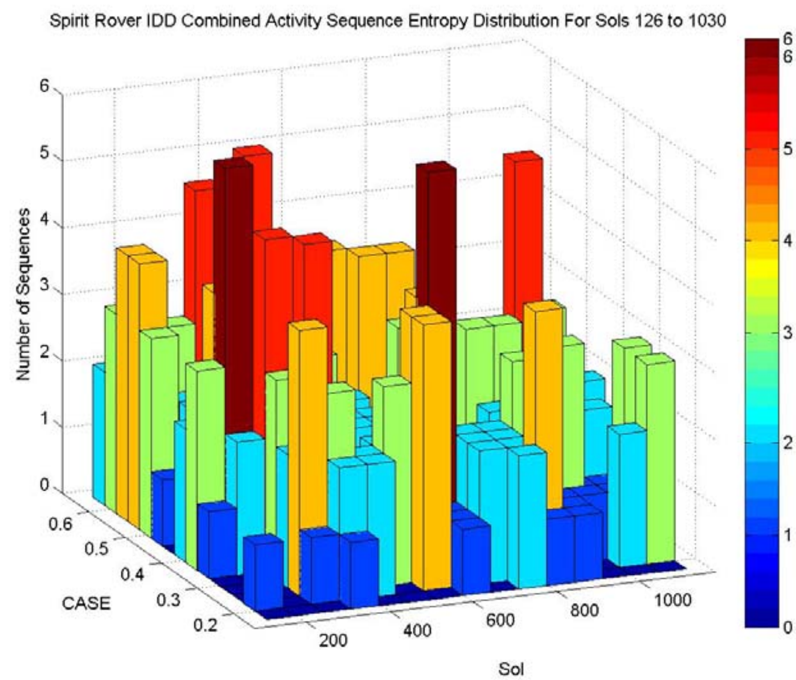

Figure 5. Spirit's IDD Combined Activity Sequence Entropy Distribution from Sol 126 to 1030

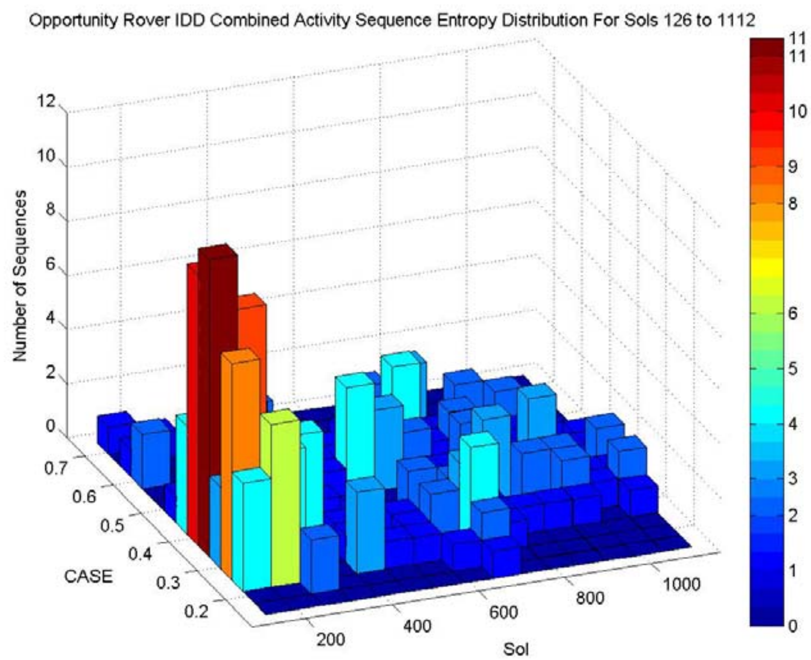

Figure 6. Opportunity's IDD Combined Activity Sequence Entropy Distribution from Sol 126 to 1112

4.3 MER IDD 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 command sequence generation. The RP team was stable with no significant turn over for the period under consideration. In addition, several of the RPs were also developers of the sequencing tools and are master users of the sequencing tools. The RP group is the most stable of the IST with very high morale and camaraderie. PACE for IDD command sequence generation was defined as follows:

$$\Omega_k = H(S_k) \times \kappa_k \qquad \text{Equation 3}$$

On MER a schedule is imposed on the command sequence generation process, in this paper the imposed schedule is expressed as a duration range, where Max_Duration and Min_Duration 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;

$$\kappa_k = \left(1 + \left[(1 + f_k) \times \left[\frac{Max\_Duration - Actual\_Duration_k}{Actual\_Duration_k - Min\_Duration}\right]\right]\right) \qquad \text{Equation 4}$$

where k is the Sol number and $f \in (0,1]$ *the plan activity* exp*erience weighting*

else κ_k is defined as;

$$\kappa_k = \left(1 + \left[(1 + f_k) \times \left[\frac{Min\_Duration - Actual\_Duration_k}{Max\_Duration - Actual\_Duration_k}\right]\right]\right) \qquad \text{Equation 5}$$

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_Duration the duration multiplier in κ_k expression is set to zero.

Min_Duration was set to 8 hours and Max_Duration was set to 10 hours, which reflects the assigned duration to generate all of the RP command sequences for both IDD and Mobility activities for a single or multiple sols. The operations team experience was set to zero (f=0) from sols 1 to 300, and 0.3 (f=0.3) from sols 301 to 1100.

Figures 7 and 8 depict Bar Graphs of Spirit and Opportunity rovers IDD **PACE** for Sols 126 to 1112. 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 will be trending upward over the lifetime of MER surface operations. Figures 7 and 8 confirm this trend and show that IDD Activity Plans for both rovers have been steadily increasing in complexity as a result of limited resources, e.g., shorter duration to generate IDD command and multiple-sol command sequences. The trend shows almost a doubling of the complexity of IDD activities plans from sol 300 onwards for both rovers.

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 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 on Mars is science return, a quantity not easily measured because it is subjective. However, it is generally understood that the MER rover's science return is not directly proportional to the number of planned IDD 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? The subjective impression of the MER project management team is that activity plan complexity has reached an upper threshold. Figures 7 and 8 confirms this with the leveling of PACE for both rovers.

Figures 9 and 10 depict a 3D histogram of Spirit and Opportunity's PACE distribution respectively From Figures 9 and 10 the two variables PACE and sol define the location in the plane, the number of sequences, the height of the corresponding column, and its color (the more intense the color the higher the PACE).

Figure 9 shows that Spirit's maximal PACE is primarily located between sols 180 to 514. As expected, this corresponds with the location for Spirit's maximal CASE. In Figure 10, a similar graph for Opportunity shows the maximal CASE located between sols 132 to 315 as expected.

Table 5 shows the summary statistics for Spirit and Opportunity IDD PACE, as would be expected except Spirit has the most complex planned activity for the period analyzed.

Table 5. Summary Statistics of IDD PACE for Spirit and Opportunity Rovers

	Summary Statistics of Spirit and Opportunity Rovers IDD PACE				
Rover	Mean	Standard Deviation	Mode	Max	Min
Spirit	0.385	0.197	0.124	0.899	0.075
Opportunity	0.355	0.219	0.104	0.874	0.069

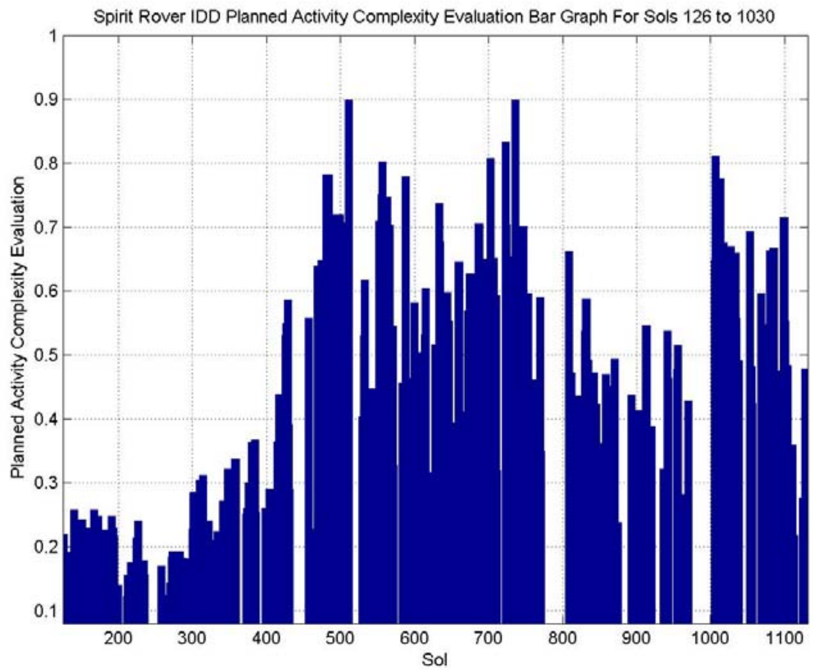

Figure 7. Spirit Rover IDD Planned Activity Complexity Evaluation from Sol 126 to 1030.

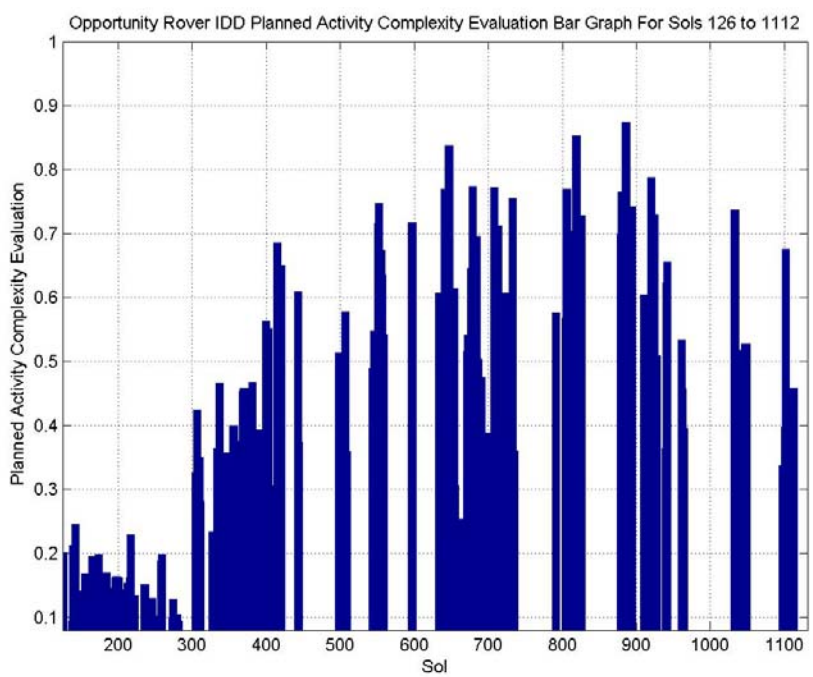

Figure 8. Opportunity Rover IDD Planned Activity Complexity Evaluation from Sol 126 to 1112.

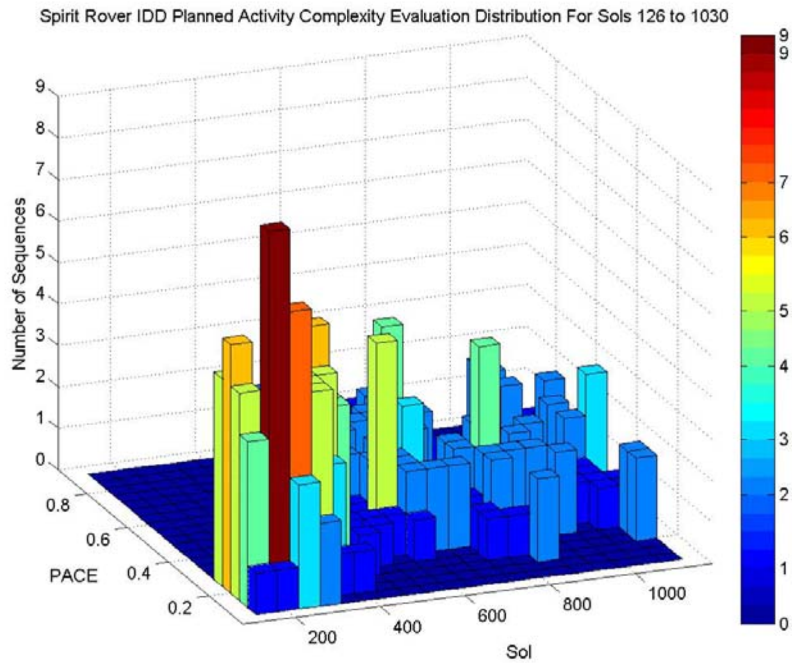

Figure 9. Spirit Rover IDD Planned Activity Complexity Evaluation Distribution from Sol 126 to 1030.

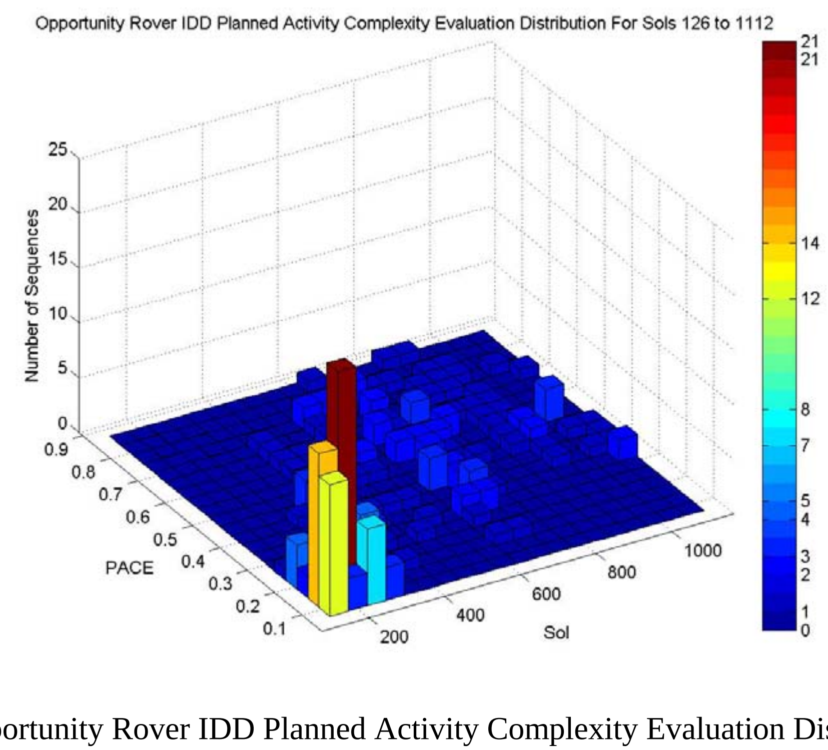

Figure 10. Opportunity Rover IDD Planned Activity Complexity Evaluation Distribution from Sol 126 to 1112.

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.

6 References

1. Mishkin, A.H.; Limonadi, D.; Laubach, S.L.; Bass, D.S.; "Working the Martian night shift - the MER surface operations process," *Robotics & Automation Magazine*, IEEE, **13**, Issue 2, June 2006 Page(s):46 – 53
2. R. McEliece. *The Theory of Information and Coding*. Cambridge University Press, New York, NY, 1984.
3. 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," in *Proc. IEEE Syst., Man, Cybern. Conf.*, 2005, **2**, pp. 1384–1388.
4. Kan Stephen H.; "Metrics and Models in Software Quality Engineering"; Addison-Wesley Publishing Company 1995.
5. Harrison Warren; "An Entropy-Based Measure of Software Complexity"; *IEEE Transactions on Software Engineering*; **18**; No.11; Nov.1992; pp.1025-1029.
6. Lind Randy K. & Vairavan K.; "An Experimental Investigation of Software Metrics and Their Relationship to Software Development Effort"; *IEEE Transactions on Software Engineering*; **15**; No.5; May.1988; pp.649-653.
7. Banker R. D., Datar S. M., Kemerer C. F. & Zweig D.; "Software Complexity and Maintenance Costs"; *Communication of the ACM*; **36**; No.11; Nov.1993; pp.81-95.
8. Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY: 1990.
9. Evans, Michael W. & Marciniak, John. Software Quality Assurance and Management. New York, NY: John Wiley & Sons, Inc., 1987.
10. C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*, Prentice Hall, 2000
11. Norris, Jeffrey S., Powell, Mark W., Vona, Marsette A., Backes, Paul G., Wick, Justin V. "Mars Exploration Rover Operations with the Science Activity Planner". *Proceedings of IEEE Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

12. A. Trebi-Ollennu, E.T. Baumgartner, P.C. Leger, and R.G. Bonitz, "Robotic arm in-situ operations for the mars exploration rovers surface mission," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Waikoloa, HI, Oct. 2005, pp. 1799–1806.

13. S. W. Squyres, et al., "The Spirit Rover's Athena Science Investigation at Gusev Crater, Mars," *Science*, 305, 794, 2004.

14. Baumgartner, E.T.; Bonitz, R.G.; Melko, J.P.; Shiraishi, L.R.; Leger, P.C.; Trebi-Ollennu, A.; "Mobile manipulation for the Mars exploration rover - a dexterous and robust instrument positioning system," *Robotics & Automation Magazine*, IEEE, **13**, Issue 2, June 2006 Page(s):27 - 36