The Unparalleled Systems Engineering of MSL's Backup Entry, Descent, and Landing System: Second Chance

Chris Roumeliotis Chris.Roumeliotis@jpl.nasa.gov[1], **Jonathan Grinblat** jonathan.f.grinblat@jpl.nasa.gov[1], **Glenn Reeves** glenn.e.reeves@jpl.nasa.gov[1]

[1]**Jet Propulsion Laboratory,** California Institute of Technology**,** 4800 Oak Grove Drive**,** Pasadena, CA 91109-8099

Abstract - *Second Chance (SECC) was a bare bones version of Mars Science Laboratory's (MSL) Entry Descent & Landing (EDL) flight software that ran on Curiosity's backup computer, which could have taken over swiftly in the event of a reset of Curiosity's prime computer, in order to land her safely on Mars. Without SECC, a reset of Curiosity's prime computer would have lead to catastrophic mission failure. Even though a reset of the prime computer never occurred, SECC had the important responsibility as EDL's guardian angel, and this responsibility would not have seen such success without unparalleled systems engineering. This paper will focus on the systems engineering behind SECC: Covering a brief overview of SECC's design, the intense schedule to use SECC as a backup system, the verification and validation of the system's "Do No Harm" mandate, the system's overall functional performance, and finally, its use on the fateful day of August 5th, 2012.*

Keywords: Mars Science Laboratory (MSL); Entry Descent and Landing (EDL); Second Chance (SECC); Fault Protection.

1 Introduction

On a basic level, Second Chance (SECC) is a version of MSL's Entry Descent and Landing (EDL) flight software that ran on the back-up flight computer. It was specifically designed so that it could have taken over and landed MSL safely on the surface of Mars in the event of a reset on the prime computer. This afforded an opportunity to partially mitigate the risk of a reset of the prime computer due to flight software or due to a non-common mode hardware reset induced during EDL. Without SECC, a reset of this nature would have ultimately been mission catastrophic. The existing MSL avionics and flight software provided the necessary infrastructure and foundation for a dual string EDL. Thus, the EDL team embarked on designing, verifying, validating, and uplinking on an accelerated schedule, installing SECC onboard MSL 24 hours before EDL was to begin. This paper will provide a very brief overview of SECC's design, along with an overview of MSL's rover compute element (RCE) for context, and then delve into the overall verification and validation (VnV) strategy of SECC. This includes Functional & Performance VnV along with "Do No Harm" VnV, SECC uplink, and SECC's final execution on the fateful day of August 5th, 2012.

1.1 SECC Design and Implementation Overview

As mentioned earlier, SECC is an extension of EDL flight software that is stripped down to the core essentials crucial for delivering the rover safely to the surface. For reference the EDL flight software image size was approximately 20MB, while the SECC image size was half that, or approximately 10MB. SECC runs on the backup computer and constantly tracks the state of EDL flight software running on the prime computer, propagating attitude and position knowledge. By gathering all the necessary data from the prime computer and bus monitor data streams, a state estimate is continuously made by SECC. This state estimate is then fed to a navigation filter, which allows SECC to be constantly shadowing the prime computer. In the event of a prime computer reset, SECC can quickly place the backup computer in control of EDL within a fraction of a second. Prior to executing EDL, all vital flight software and hardware states are configured to their last known or desired states at the time of reset. This transition of control to the backup computer is absolutely crucial, as not only all the proper hardware and software states must be configured properly and executed seamlessly, but this transition of control must be achieved rapidly enough so that the harsh physics of EDL do not become insurmountable to the vehicle. In other words, SECC had to be designed to limit the control outage so that the spacecraft could recover quickly enough to continue with EDL. SECC must provide coverage throughout all of EDL's perilous states, as shown in Figure 1.

© 2013 California Institute of Technology.
Government sponsorship acknowledged.

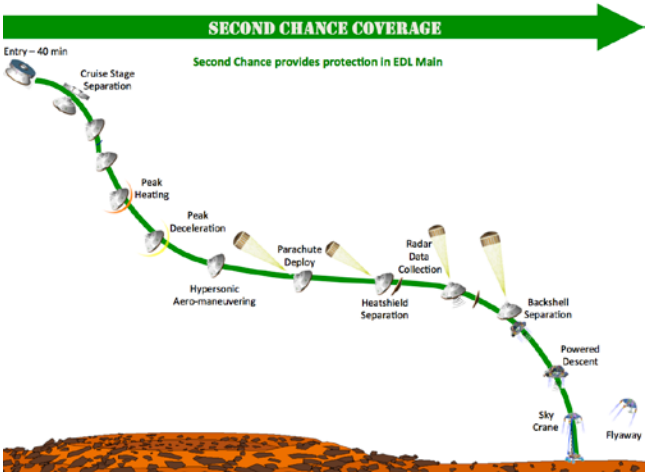

Figure 1. SECC's coverage extended through the entire EDL timeline providing protection from a prime computer reset.

Below is a simplified architecture diagram depicting the interfaces between MSL's dual rover compute elements (RCE) and MSL's myriad other essential avionics found on the rover stage, descent stage, and cruise stage. SECC had to passively run on the backup RCE (labeled as RCE-B) inflicting no interference on the prime string until there was a reset on the prime RCE (labeled as RCE-A).

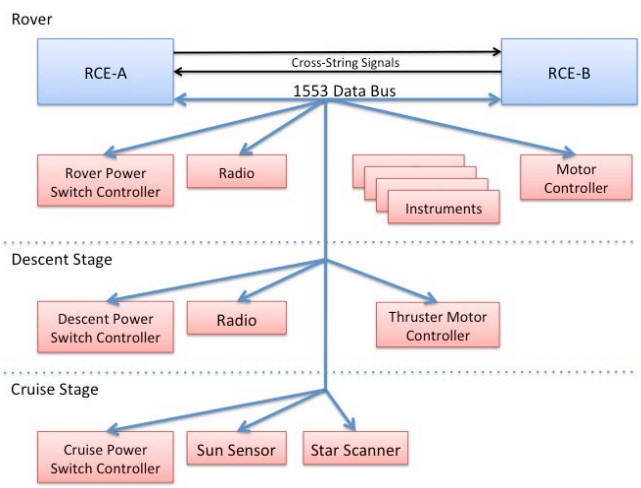

Figure 2. Simplistic diagram of MSL's avionics architecture

1.2 Rover Compute Element Overview

The Rover Compute Element (RCE) boxes perform the Command and Data Handling (CD&H) function for the entire spacecraft. As previously mentioned, there are two RCEs, that each act as a separate, redundant string. There is a primary RCE performing all the functions necessary for the entire spacecraft (cruise, descent, and rover stages), and a backup RCE, whose function depends on the software running on it, but has no control of the spacecraft. Each RCE contains a BAE RAD750 flight computer, along with custom JPL built boards to handle all the input and output (I/O) to every part of the spacecraft and its sensors and instruments. This includes radios, radar, power switch modules, inertial measurement units (IMU), sun sensor, star tracker, scientific instruments, timekeeping, etc. The RCE also is the controller of the spacecraft communications buses. The power switches to control the power for each RCE is located in a separate avionics box, which has rudimentary software that is able to autonomously power cycle or swap RCEs as needed in the event of a fault.

The main spacecraft flight software always runs on the primary RCE and is in full control of the spacecraft. The backup RCE is capable of three modes of operation. The first mode is a method of cold spare, in which the regular flight software runs on the backup computer, and only performs limited telemetry collections. If an RCE swap is needed, there can be a control outage of up to two minutes. The second mode is a second method of cold spare, in which the backup RCE is completely off. In this case, the backup RCE would be autonomously powered on and then swapped to. This would result in a maximum control outage of three minutes. The third mode of operation of the backup RCE is as a hot spare, which is when SECC runs on it. In this mode, the backup RCE follows the primary RCE, and is ready to take over spacecraft control in less than a second.

The RCEs are capable of self-arbitration of which computer is the primary and which is the backup. This is known as string arbitration, and each RCE essentially monitors the other. The primary string monitors the backup string, only to monitor the health of the backup string. The backup string monitors the primary string in order to determine if it should become the primary computer and assert control of the spacecraft. This is achieved through direct cross-string signals between the RCEs, along with software that uses these signals in order to determine which RCE should be in control. The cross-string signal names are Prime, Online, and Healthbeat. The 3 Prime signals go high on the one RCE that asserts itself primary, the Online signals go high on each RCE when software is successfully running on that RCE, and Healthbeat toggles every 125ms. Using these, software can effectively allow for only one RCE to assert control at any moment in time, and allows for an RCE to take over control as needed.

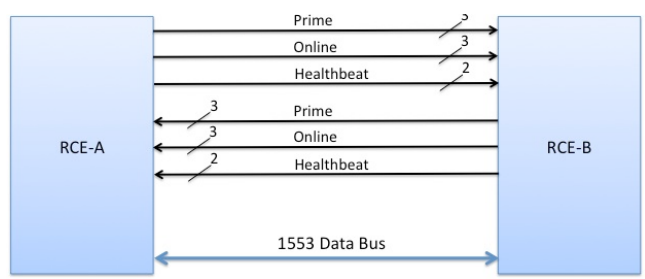

Figure 3. Diagram of interconnects between each Rover Compute Element

The ability for the backup RCE to assert control at any time was one of the most important aspects that had to be validated in the SECC software, to ensure that it only did so

at the correct time, and did not interfere with a functioning primary computer.

1.3 A Delicate System

Leading up to pre-launch, the main EDL VnV campaign baselined the backup RCE to remain powered off during EDL. In the beginning, this single string approach to EDL presented some advantages which included minimizing the chances of inadvertent interactions between the two RCEs during EDL, and more enticingly drastically reducing testing complexity. However, as the advent of the Martian landing approached, it was realized upon reassessment that relying on only one RCE left the spacecraft vulnerable to a potential reset of the prime RCE during EDL. A reset of the prime RCE was a fault case from which MSL could not recover. Over time the need for SECC became more obvious. The need for SECC was driven by MSL flight system's daunting complexity, which made the system brittle when run in single-string mode. There were many pieces of hardware and software that needed to function correctly in order to have a successful EDL.

One of the most complex parts of the system is the RCE, since any hiccup in the computer would result in the loss of the mission. Late RCE testing had caught some susceptibilities to noise on the power bus, which could result in the RCE resetting especially during pyrotechnic firings used in critical separation events. All of the plausible cases were fixed, but there was enough remaining uneasiness on what may not have been caught lurking around the corner that made a suitable backup highly desirable. It was decided that the usage of the redundant RCE would add indispensible robustness to the system.

2 Overall Verification and Validation Strategy

The underlying challenge was to ensure SECC could be armed and allowed to take control, but at the same time not interfere with EDL. In other words, SECC had to functionally perform exactly like EDL, but only when required to do so. If SECC was not needed, i.e. there was no reset on the prime string, SECC had to "Do No Harm" to the prime string and the rest of the flight system, while passively running on the backup string. Since all previous EDL testing had occurred with the backup RCE off, including simulated EDL on the real flight spacecraft, the SECC verification and validation program provided unique challenges. It would never get to run on the real spacecraft until the real EDL, so it was necessary to prove that it was providing insurance, without increasing risk to a successful landing.

Additionally, the work on SECC only began less than a year before landing. This meant that the normal verification and validation program that can take years to complete would not suffice. Thus, an extremely thorough VnV strategy was developed that ensured all aspects of the system were tested in the most robust manner. Not only was the functional and performance side tested but also the "Do No Harm" side. Below is a diagram depicting the overall roadmap of the SECC VnV strategy. The VnV strategy evolved into a dual pronged approach with an emphasis on the "Do No Harm" paradigm feeding into both prongs. This "Do No Harm" paradigm needed to permeate throughout the entire VnV process, as this was absolutely essential in driving and molding the VnV scope as a whole.

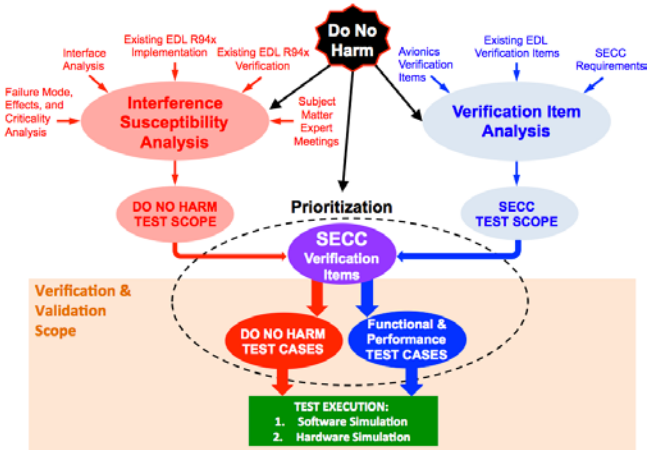

Figure 4. SECC's VnV strategy roadmap consisted of a dual pronged approach.

One prong concentrated on developing the "Do No Harm" test scope through a separate means of an extremely detailed interference susceptibly analysis. The other prong concentrated on developing the SECC functional and performance test scope through a means of verification item analysis. The diagram visually depicts the dual pronged approach for SECC's VnV strategy with both prongs merging downstream to create the entire VnV scope.

Different input products and analysis flowed down through each prong creating a separate test scope, generating a prioritized verification item (VI) superset, finally breaking these down further into test cases on different testbed venues. The interference susceptibility analysis prong produced 24 "Do No Harm" VIs while the verification item analysis prong produced 322 "Functional & Performance" VIs. Among the superset of VIs, the "Do No Harm" VIs resulting from the analysis and flow down were prioritized the highest, as they were the most essential in verifying and validating.

Figure 5 depicts the verification item burn-down as a function of schedule. The burn-down of the "Do No Harm" (DNH) VIs were assigned the highest priority and are depicted in red, while the "Functional/Performance" VIs were completed subsequently and are depicted in blue. All VIs needed to be burned down to embark on Certify for Use Testing to further justify SECCs use in flight. This only allowed for 3 months of rigorous testing. In the beginning, all focus was kept on the "Do No Harm" aspect until we felt

completely comfortable and convinced proceeding with the burn-down of the "Functional/Performance" aspect. This meant that very close attention needed to be given so that we did not miss anything in the "Do No Harm" VnV scope, to ensure we did not misappropriate critical resources that were to be released to bolster the burn-down of the "Functional/Performance" aspect. All testing needed to be complete before the Uplink Readiness milestone. This burn-down occurred in parallel with various SECC builds, which would fix the issues found during the test campaign. This allowed the team to release new SECC builds strategically within the VnV timeline marked in the green diamonds below. All builds needed to be regression tested as well to ensure that the correct changes were applied. This meant that the new builds injected within the SECC VnV timeline needed to have extra testing done to re-verify and re-validate all the test scope that had already been done previously.

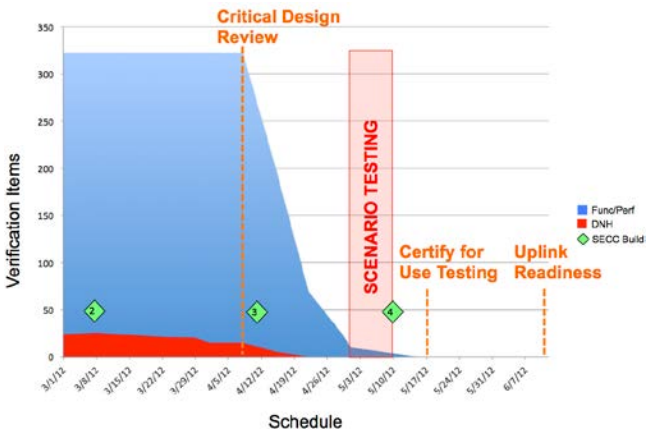

Figure 5. SECC's verification item burn-down superimposed on a schedule with crucial milestones.

2.1 Functional & Performance Verification and Validation

From the total 322 "Functional/Performance" VIs, different test cases needed to be concocted to strain the system and verify that SECC would perform as designed. If a reset were to occur, SECC had to land the vehicle safely on the Martian surface. For SECC, the EDL success criteria were slightly relaxed to remove some of the built-in conservatism that was originally placed on the main EDL flight software. If a reset occurred and SECC took over the vehicle, then we did not want to unnecessarily constrain SECC as much as the main EDL flight software as this would end up biasing our perception and lead us to falsely rate SECC's true performance. Some of the relaxed success criteria included slightly higher vertical and horizontal touchdown velocities, more than one but less than three touchdown events during the sky crane maneuver before flyaway, and slightly larger landing ellipse positional errors.

An example of a test was to induce a reset on the prime string within the terminal descent radar data collection portion of the EDL timeline, and verify that SECC could land the vehicle safely. A reset of the prime string occurring during this portion of the timeline was deemed one of the most plausible. Though the test was was difficult to get right it was accurately shown that, SECC was passively monitoring the state of EDL up until the prime string reset. At the time of the reset, SECC kick-started and took over the entire vehicle within a short 0.71 seconds, flawlessly putting the vehicle in a heading alignment slew as expected. Had the control outage been larger the landing would not have been successful, as it would have failed most of the success criteria. However, the 0.71s control outage was small enough to declare success, resulting in a landing that was a little farther downrange from the target but with better than expected touchdown velocities. This was only one example, but during the full gamut of the testing, the team began developing more and more confidence with SECC's performance. Through this testing, it could be demonstrated that SECC was able to land the vehicle with a prime string reset at almost any point on the entire EDL timeline.

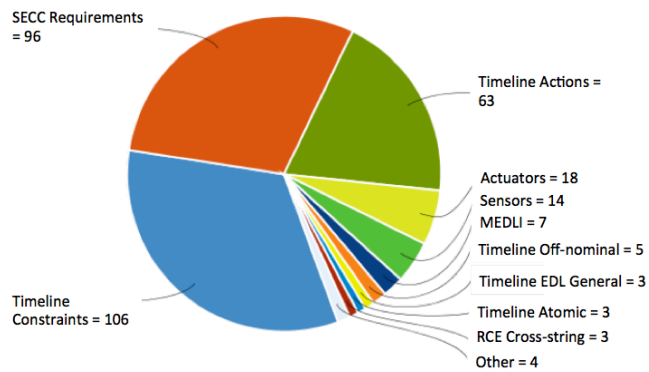

Figure 6. A pie chart of SECC's functional and performance VI areas.

To give a sense of breadth of the VIs that were covered to ensure SECC was functioning as designed, the above pie chart details the breakdown all of SECC's "Functional/Performance" VIs. The functional and performance VIs that needed to be verified ranged from pyro firings within the EDL timeline constraint area (highlighted light blue in the figure), all the way to SECC performance augmentation requirements found within the SECC requirements area (highlighted dark orange in the figure). These VIs were all burnt down in time for Certify for Use testing.

2.2 Do No Harm Verification and Validation

In order to upload and enable the SECC software on the backup RCE, it was important to make sure that it never tried to assert control of the spacecraft when it wasn't supposed to. If there were no faults during EDL, then SECC is never supposed to try to take over, which is what was called "Do No Harm." Verification and validation of this requirement was top priority, since if SECC did not work at

all, then at least MSL would not be any worse off than it was before. As previously mentioned, each RCE has a software module that controls if it is allowed to become the primary RCE and take control of the spacecraft. This software module is called string arbitration (SARB). Most of this SARB software remained unchanged when imported for use with SECC, but due to the criticality of this piece of software, a full VnV program was warranted for it.

In order to verify that SECC worked correctly in the nominal case and was never needed, the focus was put onto three main mission phases. First, when enabled on the backup computer during pre-Entry Descent and Landing operations (pre-EDL), it should have no effect on the operation of the spacecraft, and from the primary computer's perspective, nothing on the flight system changed. The second phase of focus was during Entry and Descent. In this phase, again the primary computer should behave no differently if SECC was running or not, and if the backup computer was on or not. The last area of focus was Landing, in which SECC is supposed to self-remove itself and allow normal flight software to run on the backup computer which allowed the rover to safely phone home.

All verification and validation was performed on a high fidelity testbed that included nearly identical copies of all of the hardware flying to Mars. The Verification and Validation of "Do No Harm" first focused on noise on the cross-string signals. These signals are triplicate voted in software, so if any single signal has a fault, software is supposed to vote the best two out of three. There are eight signals (three unique) that go in each direction between RCE, and every combination of single signal faults were tested to verify that the backup RCE running SECC never incorrectly tried to take over. Next, the focus was put on if the backup computer went belly-up during EDL, since all previous EDL testing had been done with the backup RCE off. During multiple simulated runs of pre-EDL an EDL, the backup RCE was purposefully reset over and over, which the primary computer is supposed to just ignore, aside from just printing a message. The last area that was tested was to make sure that SECC correctly removed itself from memory and left no trace once Curiosity landed successfully. This was done with memory readouts before and after simulated landings to verify that nothing had changed before and after SECC was run.

As a result of the tight schedule in the implementation plan for SECC, it was important to streamline the verification and validation to allow for regression testing. There were a number of successive releases of the software as functionality was added and bugs were fixed. This required coming up with a limited regression test that was used between minor releases, which were separated by one to two weeks. This limited test could be completed in one 8-hour shift on the hardware testbed. For release candidate builds, the entire "Do No Harm" verification could be completed in two to three 8-hour shifts. With the streamlined regression plan, it was possible to quickly turn around testing after a release in order to allow for the functional testing of SECC recovering EDL under various scenarios to continue.

3 Uplink and Final Execution

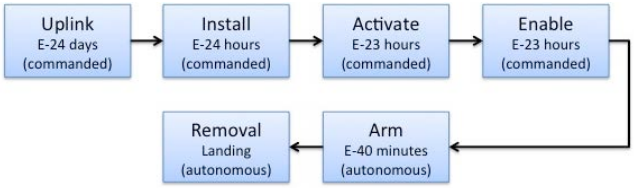

Figure 7. Flow diagram of running SECC on the flight spacecraft

Before SECC could be uplinked to the spacecraft for use, SECC underwent an additional "Certify for Use" test campaign. After all the verification items were burned down, the "Certify for Use" campaign began with the final build of SECC. This campaign involved targeted scenario testing which re-tested the most likely reset scenarios, performed a complete regression test campaign of the "Do No Harm" aspect of the VnV, and involved two flight-like operation readiness tests (ORTs). In order to verify that SECC could be loaded onto the flight vehicle in a flight-like manner, the exact procedure that was to be used 24 hours before EDL was performed during these two ORTs. ORTs are special tests in which the MSL testbed is set up to match the flight vehicle as closely as possible. The flight operations team then runs through EDL on the testbed as if it is the real vehicle. All of the procedures and commands that would be sent in real life in the lead up to EDL, are sent here as well. This ensures that the ordering of the commands has no unexpected effect, and that the flight operations team can perform all the necessary tasks in the allotted time. During these two ORTs, in which SECC was loaded, it was demonstrated that SECC could be installed and run on the flight vehicle with no issues.

When the time came for the real EDL preparations, the SECC load went exactly as planned, with the results being no different from what was seen during the ORTs. The SECC software was uplinked about a month before landing. The uplink process first involved compressing the software image and then dividing into twenty-two small files. The files were then uplinked via the Deep Space Network (DSN) over the course of two days. Finally, once confirmation that all of the files were on board was received, the files were re-merged back together, the image was decompressed, and a checksum was performed. This SECC image then sat on the backup computer for the next month until it was time to install and activate it.

About 24 hours before EDL, the SECC software was installed on the backup RCE, which involved three main steps. The first step was copying it into volatile RAM. Loading the SECC software image in volatile memory instead of non-volatile memory provided one major benefit: should the backup RCE lose power for any reason, regular

flight software would then load on it when power is restored. This was part of the "Do No Harm" strategy. The second step was activating SECC by resetting the backup computer, which allowed it to boot into the SECC software image. Lastly, SECC was enabled via a ground command sent to the primary RCE, which gave the go-ahead to the backup computer to allow SECC to take over as needed in the event of a fault after forty minutes before entry (E-40).

At E-40, the SECC software autonomously armed itself as designed, and all telemetry indicated that it was perfectly following the events initiated by the primary computer. Throughout entry, descent, and landing, the telemetry all indicated that the SECC was working and could have taken over spacecraft control at any time.

Once the landing sequence completed, and the rover was safely on the ground, the SECC software successfully removed itself from memory and reset the backup RCE it was running on. Normal flight software then loaded on that RCE, and the remaining sequences to get ready for surface operations over the next few hours all worked nominally. The SECC software had fulfilled its goal of doing no harm, and EDL was successful. Ultimately the insurance SECC yielded was priceless to the team. Just knowing that there was a guardian angel waiting to take control in the event of a reset of the prime computer was an amazing feat for the MSL mission and the team.

4 Conclusions

SECC, from concept to execution, required precise and focused systems engineering. It required dealing with an extremely tight schedule, a limited team, the inability to test it on flight hardware before launch, and the requirement of not making the chances of a successful Entry, Descent, and Landing any lower. This posed unique challenges that were met with a unique plan of attack.

The SECC design and implementation were planned around the restricted schedule from the beginning. Having the hard deadline of August 5th, 2012 provided all the motivation needed to get done on time. By focusing on the most brittle parts of the system, it was possible to gain a high amount of confidence in the design and testing. The two-pronged approach used for verification and validation was instrumental in identifying all the test cases required to feel confident in the system's ability to perform. Lastly, focusing on the "Do No Harm" set of verification items separately from the functional test cases, allowed for easy delineation in the test program, and for quick regression testing of new builds of the software.

Although SECC was not utilized in the landing of the Curiosity rover, its development and testing allowed for the entire EDL team to gain a better understanding of how the flight system works during stressful conditions. The extensive testing of SECC did uncover a small amount of potential mission ending bugs in the mainline flight software running on the primary computer, which could be fixed before EDL. The primary computer did end up working flawlessly throughout the entire entry, descent, and landing sequence. The extra insurance SECC provided for a successful EDL, warranted all of the intense effort put it into it. SECC ingeniously used all the redundancy built into the system and maximized the chances for landing. SECC ended up setting precedence as the first "dual string" EDL system employed on a Mars mission. All previous Mars missions had used a "single string" approach before MSL. In this light, SECC paved a new way of guaranteeing a successful landing on Mars. And to this end, it will not be surprising to see the Jet Propulsion Laboratory use a very similar backup EDL system on all future Mars missions. SECC was designed and tested to truly give MSL a "second chance" at EDL, even though in the end it was not utilized. On August 5th, 2012, Curiosity's successful landing was a tribute to the hard work of everyone who had worked on the project over the preceding 9 years.

References

[1] R. Prakash, P.D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, A. M. San Martin, S. W. Sell, A. D. Steltzner, D. W. Way, "Mars Science Laboratory Entry, Descent, and Landing System Overview," 2008 IEEE Aerospace Conference, pp. 1-18, March 2008.

[2] R. P. Kornfeld, R. Prakash, A. Chen, A. S. Devereaux, M. E. Greco, C. C. Harmon, D. M. Kipp, A. M. San Martin, S. W. Sell, A. D. Steltzner, "Verification and Validation of the Mars Science Laboratory/Curiosity Rover Entry Descent and Landing System," AAS/AIAA Flight Mechanics Conference, pp. 1-30, February 2013.

Acknowledgements

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.