JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 41, No. 4, July–August 2004

Validation and Verification of Deep-Space Missions

Riley M. Duren*
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

The growing technical complexity of and increased cost pressure on deep-space science missions poses challenges for the system engineering discipline of mission validation and verification (V&V). In the wake of several recent mission failures, the aerospace community is still searching for a stable middle ground between the highly reliable yet cost-prohibitive approaches of the past and cheaper but excessively risky methods of project implementation. The key components of a robust mission level V&V program are presented, with attention on the guiding principles and considerations. The concept that a V&V program should be "three dimensional" is discussed, for example, the end-to-end aspect or width, the top-to-bottom aspect or depth, and the project life-cycle aspect or time. Distinctions are made between definition-phase requirements validation and implementation-phase system validation and verification. The role of modeling and simulation relative to system testing and the necessity of model validation are described. Validation of system robustness using techniques such as performance sensitivity analysis, system fault tree analysis, probabilistic risk analysis, and stress testing is explored. Finally, a summary V&V checklist is provided.

Introduction

IN 1997, the same year NASA's $3 billion Cassini spacecraft began its long voyage to Saturn, a fundamental change in the approach for future space-science missions was heralded with the successful landing of the $165 million Mars Pathfinder mission. NASA's near-term (2003–2013) deep-space exploration program encompasses many new missions in various stages of planning or development, the most mature of which are shown in Table 1. This suite of projects includes the relatively inexpensive (~$300 million) Discovery and Mars Scout missions, the moderately priced (~$650 million) New Frontiers missions, and a unique set of larger ($1000 million plus) missions associated with the Origins and Solar System Exploration Programs.[1] A dozen or so other deep-space missions are also being seriously considered or under development in the United States and Europe. [Note that although the focus of this paper is on deep-space missions (thus, the list of upcoming missions excludes a large collection of Earth orbiters), many of the concepts presented here are applicable to other space platforms and complex systems in general.] With the proliferation of such missions, the community of engineers, scientists, and managers delivering them is still adapting to the new mode of project implementation. There is a need to find some stable middle ground between the very reliable but cost-prohibitive methods employed on Cassini, Galileo, and Voyager and the lower-cost missions currently in the queue. Designers are frequently forced away from costly redundant architectures and intensive ground-in-the-loop operations to less fault-tolerant, single-string architectures with significant onboard autonomy. Hence, the need for robustness (and proof of such robustness) has increased for hardware, software, and operational processes. Recent attempts to implement missions with the faster, better, cheaper (FBC) approach have suffered from mixed results. On the one hand, the Mars Pathfinder, Lunar Prospector, Near Earth Asteroid Rendezvous (NEAR), and Deep Space 1 (DS1) missions were successfully implemented in the FBC mode. Genesis and StarDust continue to operate nominally after several years in space. However, those successes have been tempered by catastrophic failures of the Mars Climate Orbiter (MCO), Mars Polar Lander (MPL), Wide-Field Infrared Explorer (WIRE), and Comet Nucleus Tour (CONTOUR) missions. This mixed track record encourages continued refinement of our project implementation practices to emphasize success first and, in particular, the validation and verification (V&V) techniques needed to ensure system robustness critical on missions with relatively small teams in which every link in the risk-avoidance chain must be protected. One should also remember the lessons of the Hubble Space Telescope (HST) and Mars Observer (MO), which proved that high cost does not always correlate with reduced risk. Lessons-learned reports for such missions repeatedly highlight the paramount importance of a thorough end-to-end V&V program when trying to balance risk and cost.

Increasing complexity of the missions being launched represents another important driver for V&V. The upcoming suite of deep-space missions requires dramatic new technologies in many cases. In the space-astronomy arena, missions are driving the state of the art in many areas: Kepler's few parts-per-million photometry, the Space Interferometry Mission (SIM)'s few microarcsecond astrometry, the Laser Interferometric Space Antenna (LISA)'s fempto-G acceleration sensing, and the Terrestrial Planet Finder (TPF)'s ultrahigh contrast (10^{-10}) imaging needs are all setting new targets for performance. Getting new technologies such as picometer metrology, subangstrom optical wavefront control, and separated-spacecraft, cryogenic nulling interferometry to work on Earth is difficult enough. The V&V challenges associated with converting these to reliable, space-borne systems are formidable. Likewise, as the scope of NASA's Solar System Exploration Program expands, future missions to Mars and the outer planets drive the need for highly autonomous spacecraft remote agents, real-time hazard sensing and avoidance in planetary landers, advanced propulsion and energy systems, and the ability to operate in extremely high-radiation environments. Many of the listed capabilities are cost prohibitive or impractical to test in an end-to-end fashion before launch, thus, placing an increased burden on modeling and simulation as part of a robust V&V program. There is also a caution here: Projects faced with daunting technological challenges often succumb to tunnel vision and focus on invention, while neglecting the more mundane aspects of the system, such as the spacecraft bus. Flight in deep space is still far from routine. Projects must continue to apply significant attention to basic health and safety issues, such as fault tolerance and fault protection design and validation.

V&V: Definitions and Scope

Although independent V&V (IV&V) of software has received considerable attention recently, standards for the broader mission-level V&V activity are rather fuzzy [despite attempts by the International Organization for Standardization (ISO) and other

Received 3 April 2003; revision received 11 August 2003; accepted for publication 12 August 2003. Copyright © 2003 by the American Institute of Aeronautics and Astronautics, Inc. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0022-4650/04 $10.00 in correspondence with the CCC.

*Senior Project Engineer, Mission and Systems Architecture Section, 4800 Oak Grove Drive, Mail Stop 301-450.

Table 1 NASA deep-space missions 2003–2013

Mission	Launch
Space Infrared Telescope Facility (SIRTF)	2003
MER	2003
Deep Impact	2004
MESSENGER	2004
STEREO	2005
MRO	2005
Dawn	2006
Kepler	2007
Mars Scout 1–2 (2)	2007–2011
Discovery 11–13 (3)	2007–2011
New Frontiers 1–2 (2)	2009–2011
Mars Science Laboratory (MSL)	2009
Mars Telecom Orbiter	2009
SIM	2009
James Webb Space Telescope (JWST)	2010
Jovian Icy Moon Orbiter (JIMO)	2011
LISA	2012
TPF	2013
Mars Sample Return (MSR)	2013

organizations], other than the mundane aspects of verification.[2] Likewise, the scale of mission-level V&V efforts and basic approach varies widely from project-to-project. In fact, there appears to be a cultural aspect of how V&V, and validation in particular, is treated in various organizations. Namely, "great system houses" are often adept at using validation to catch problems in the design phase, whereas "great integration houses" have demonstrated an ability to find creative solutions for horrendous problems that arise late in the implementation phase.[3] Missions have been successfully implemented by both cultures, but the latter typically suffers from increased risk and higher costs. (It is cheaper to correct problems earlier.)

The following definitions are provided to illustrate the subtle but important distinctions between different aspects of a V&V program. The commonly used phrase, "Verification proves the design is right; validation proves it is the right design,"[4] is correct but not very specific. The author has used a combination of sources including the handbooks from NASA[5] and ISO and personal experience to refine the common definitions as follows. Verification is proof of compliance of the as-delivered system with specific requirements, that is, does what we built meet the requirements we wrote? Whereas proving compliance with top-level performance requirements can sometimes be a challenging task, the basic concepts behind verification are fairly straightforward and universally recognized, such as the use of verification matrices. Validation is a more nebulous concept and can be broken into three subdefinitions: requirements validation, model validation, and system validation. Requirements validation is proof that the requirements (and, hence, the system design) should satisfy the customer's need or purpose before the system is actually built. Model validation is proof that the models and simulations to be used for requirements and system validation are correct. System validation is proof that the as-delivered system (all project elements operating end-to-end in the expected flight environment with reasonable stressing conditions) will meet the driving need, that is, does what we built meet the objectives? It is important to recognize these distinctions between verification and validation.

The structure of a generic V&V program is shown in Fig. 1. The driving need is used to set the initial requirements and is frequently referred back to as part of the validation activity. Initial model validation is done to ensure the models/simulation can be used safely to support requirements validation. In addition to performance analysis/modeling/simulation, requirements validation includes risk analysis such as fault tree analyses and probabilistic risk analyses to ensure the design will be robust against failures or off-nominal situations. This will be discussed further in upcoming sections. V&V requirements and a set of tracking matrices are used to cover requirements validation, model validation, verification, and system validation.

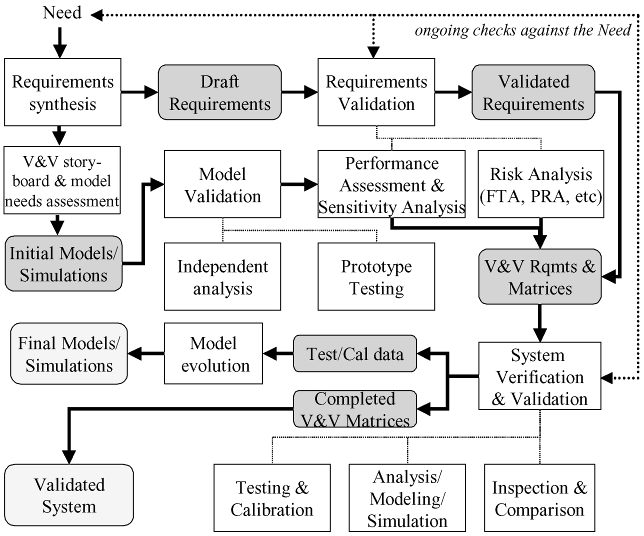

Fig. 1 Generic V&V program structure.

System-level V&V, including rollup from lower-level V&V, consist of testing and calibration (characterization) of the as-built system, analysis, modeling; simulation of untestable aspects; and inspection for others. The output of the V&V activity includes updates to the V&V matrices and test/calibration data for updating the system models and simulations (for potential future use in flight). When the V&V matrices are complete, the system can finally be considered validated (should satisfy the need). Of course, a system is not truly 100% validated until the mission is successfully operational, but every project must strive to achieve a reasonable confidence level before launch. The definition of reasonable varies from project to project although most strive for >90% confidence at launch.

A good V&V program is three dimensional in nature. (Each dimension is shown in Figs. 2–4.) There is an end-to-end aspect of the mission that spans its entire functional and performance space, which can be thought of dimensionally as width. For an astronomy mission this would consist of considering everything from photons incident at the aperture (including predicted characteristics of the source) through filtering and detection by the flight instrument(s), data storage and transmission by the spacecraft and deep-space network, error correction and data reduction by the science data center, and analysis/interpretation by the science team, as well as the operational procedures used to implement the described process. Figure 2 provides a high-level snap-shot of this example (for the primary data flowpath).

The second dimension of a V&V program is the top-to-bottom aspect or depth. The familiar V model in systems engineering (Fig. 3) reflects the hierarchical structure of the mission and the process by which requirements are decomposed into suballocations and validated on the downstroke and then the design and as-built system are verified and validated on the upstroke.[3] Whereas verification is typically performed at all levels, validation in its most rigorous and comprehensive form occurs at the system level. However, in a well-executed project, some validation is performed at all levels. For example, a designer of an electronic board will typically perform a worst-case analysis to assess the design robustness to stressing conditions. Such lower level validation activities are flowed upward to the next level, playing a part in overall system validation.

The third dimension of a V&V program is time. The V&V program must span the entire project cycle, from formulation, definition, development, test, operations, and data reduction (phases A–E in NASA parlance). As shown in Fig. 4, the V&V program evolves over the project life cycle. It is also critical that validation address the full mission duration space, covering time-dependent factors and all scenarios. Note that although uplink command validation and software-patch validation are important parts of operations, they are considered peripheral to this discussion.

The various V&V activities will be explored further in upcoming sections, but first some comments about balancing cost vs risk are

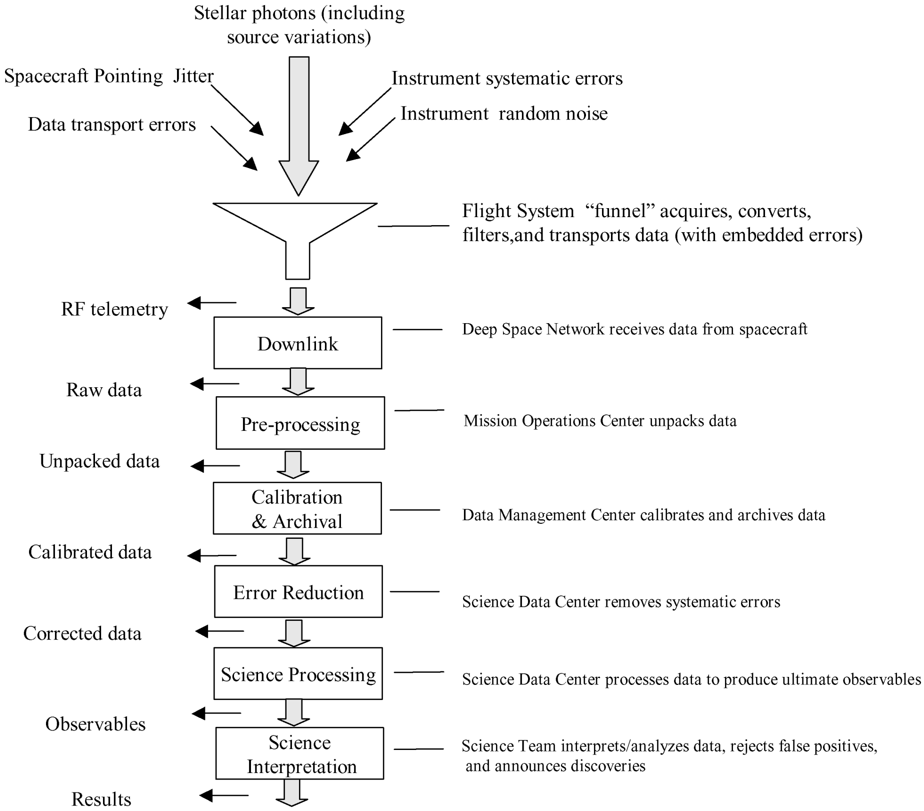
Stellar photons (including source variations)
Spacecraft Pointing Jitter
Data transport errors
Instrument systematic errors
Instrument random noise
Flight System "funnel" acquires, converts, filters,and transports data (with embedded errors)
RF telemetry
Downlink
Deep Space Network receives data from spacecraft
Raw data
Pre-processing
Mission Operations Center unpacks data
Unpacked data
Calibration & Archival
Data Management Center calibrates and archives data
Calibrated data
Error Reduction
Science Data Center removes systematic errors
Corrected data
Science Processing
Science Data Center processes data to produce ultimate observables
Observables
Science Interpretation
Science Team interprets/analyzes data, rejects false positives, and announces discoveries
Results

Fig. 2 End-to-end functional data flow (width).

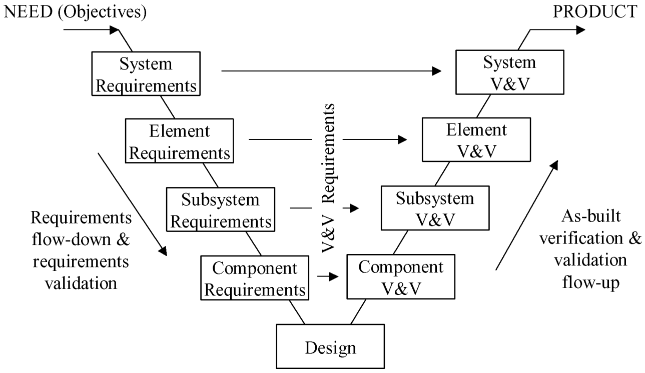

Fig. 3 Systems engineering top-to-bottom V model (depth).

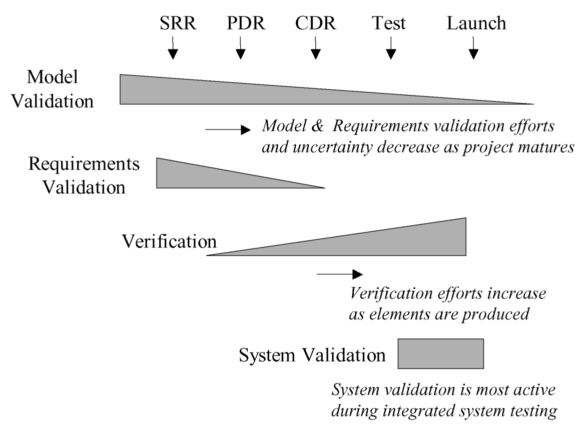
Phase: B (definition) C/D (design & build) E (operations)
SRR
PDR
CDR
Test
Launch
Model Validation
Model & Requirements validation efforts and uncertainty decrease as project matures
Requirements Validation
Verification
Verification efforts increase as elements are produced
System Validation
System validation is most active during integrated system testing

Fig. 4 V&V across the project life cycle (time).

warranted. If the mission fails, we probably did not spend enough effort on V&V. On the other hand, one can also bankrupt a project by being overly conservative. Clearly the risk posture varies from project to project and, hence, the percentage of total project cost appropriate for allocation to the V&V program. NASA's software IV&V program addresses this by designating which areas to apply formal V&V. Risk scores are estimated for each major software entity. (Risk is defined as the combination of impact and probability of occurrence.[6]) For mission-level V&V, that risk rating system can be adapted to use the following factors when assessing the risk for each area: 1) size of project organization, 2) complexity of project organization, 3) schedule pressure, 4) maturity of system engineering process on the project, 5) degree of technical innovation needed, 6) degree of system integration needed, 7) maturity of system requirements, and 8) amount of inheritance.

For example, a mission with a single, fairly self-contained instrument (in terms of its ability to meet performance requirements without the help of external entities) is easier to design, simulate, and test than one with many, tightly coupled or distributed instruments and spacecraft subsystems and a complex ground data processing and error correction function. Also, the amount of available inheritance in a particular mission element has some bearing on the scope of V&V for that element. For example, if a certain model of gyroscope has repeatedly demonstrated flight worthiness in terms of performance and robustness, in conditions similar to the new application, the gyro can be validated/verified by similarity and/or analysis, thus, saving cost on testing for that unit (although it still needs to be factored into system-level V&V effort). The important point here is that the project must go through the exercise of drafting a V&V plan early in definition phase and explore the listed topics. Considering and agreeing on the overall risk posture and corresponding scope of the V&V program is essential to achieving the proper balance.

Requirements Validation: Approach and Techniques

In the process of proving the system will meet the ultimate need, validation must address several aspects of the requirements and system design: correctness, completeness, achievability, verifiability, and robustness. Clearly, the requirements governing the system design must be correct and complete to satisfy the need. The requirements must likewise result in a design that is achievable given the allocated project resources. To avoid unreasonable risk, requirements must be such that they can be verified once the system is built. Finally, the requirements should result in a system that is both robust to variations in performance beyond the nominal operating range as well as robust in the presence of reasonable fault conditions. These points were driven home with lessons learned in the wake of the MCO and MPL failures, namely, verification such as a test-like-you-fly philosophy can never relieve engineers of their top priority: to get it right the first time.[7] Specific recommendations in this area included the following:

1) Avoid stating requirements in a negative sense (shall not do X) because they are notoriously difficult to verify.

2) Ensure that lower-level, for example, subsystem, requirements are covered properly by a parent requirement at the system level because the latter are generally subject to more rigorous, formal V&V and, thus, more likely to catch problems during later verification.

Given that V&V is a three-dimensional problem, it is important to study the driving requirements from multiple vantage points to avoid missing something critical. The following tools and techniques are available for validation, each offering a unique and important viewpoint: 1) functional flow diagrams; 2) performance error budgets; 3) performance sensitivity analyses/models, including a) merit functions and b) Monte Carlo simulations; 4) risk analyses, including a) fault tree analysis, b) probabilistic risk analysis, c) failure modes, effects, and criticality analysis, and d) worst-case analysis; and 5) early hardware/software testing.

Functional Flow Diagrams

End-to-end system functionality (the width dimension of the V&V program) can be studied with functional flow diagrams as shown in Fig. 2. The exercise of understanding the logical flow between project elements from frontend to backend offers insight into adequacy of interface requirements and overall system utility. In addition to addressing hardware/software aspects, functional flow diagrams are also important for studying operational processes and the requirements placed on them, particularly for human–machine interactions and fault response.

Performance Error Budgets

Likewise, comprehensive error budgets are useful in understanding the top-to-bottom flow of performance requirements and capabilities (addressing the depth dimension of V&V). A simplified example of an error budget from the Kepler planet-detection mission is shown in Fig. 5. The driving need/science objective and resulting top-level system requirements are highlighted. Error budgets also represent an example of the temporal aspect of validation. Early in the project, engineers use error budgets in a top-down mode to suballocate performance requirements to the different elements. Then, as the project progresses and design maturity improves, the error budgets are used in a bottom-up mode to predict the expected performance. Improving the fidelity of the error budgets in this fashion is one goal of the model validation effort.

A potential pitfall to watch for in error budgets is the appropriate treatment of systematic errors vs uncorrelated, random noise. This topic is beyond the scope of this paper, but the performance analyst must be scrupulous in understanding and accounting for the effects of systematic errors and the calibration and error rejection techniques employed in data-reduction. Carefully accounting for small errors and residuals are necessary to avoid being either over-conservative or too optimistic in estimating overall system performance. System engineers should demand that performance error budgets explicitly depict all potential error sources. (Even if the magnitude of the error contribution is negligible, the exercise must be completed.)

Robustness and Graceful Degradation

Projects sometimes make the mistake of unintentionally creating requirements and designs that result in operation at or near "cliffs." Whereas a system may be designed to meet performance specifications within a fairly tight set of tolerances around a required central value, it may fail precipitously in the event of relatively minor excursions from the region of nominal operation. The concept of graceful degradation is key to successfully implementing deep-space systems in the FBC environment. Properly executed validation programs enable graceful degradation by using performance sensitivity analyses and design risk analyses to identify cliffs and soft-spots, thus, providing the project sufficient insight to guide risk vs cost (mitigating action) trades.

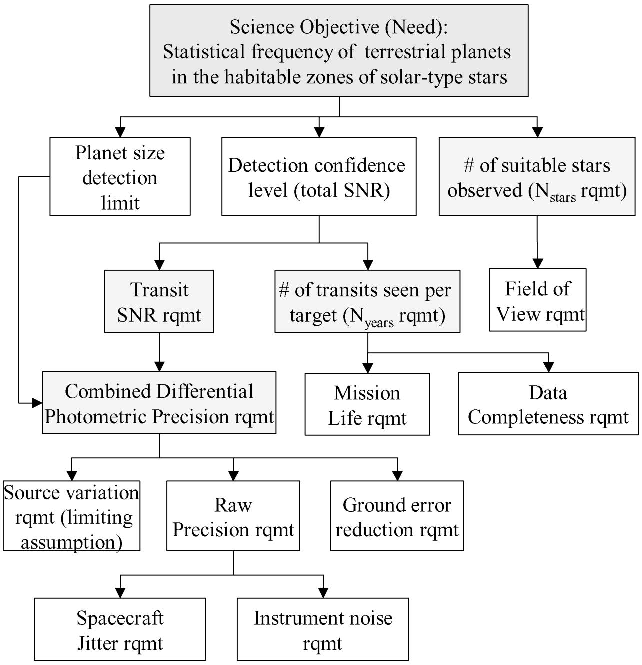
Science Objective (Need): Statistical frequency of terrestrial planets in the habitable zones of solar-type stars
Planet size detection limit
Detection confidence level (total SNR)
# of suitable stars observed (N_{stars} rqmt)
Transit SNR rqmt
# of transits seen per target (N_{years} rqmt)
Field of View rqmt
Combined Differential Photometric Precision rqmt
Mission Life rqmt
Data Completeness rqmt
Source variation rqmt (limiting assumption)
Raw Precision rqmt
Ground error reduction rqmt
Spacecraft Jitter rqmt
Instrument noise rqmt

Fig. 5 Performance error budget for Kepler.

Performance Sensitivity Analyses

Merit functions and Monte Carlo simulations are two useful tools for assessing the robustness of a system in terms of its overall performance. Monte Carlo simulations are well described in the literature and can be very helpful for perturbing system parameters such as spacecraft pointing jitter in stressing cases to assess the impact on top-level performance.[5]

Merit functions are models that provide the system engineer and the customer [such as a principal investigator (PI)] with a method of studying the sensitivity of the mission need to changes in key mission parameters. Another way of thinking about this is what is the science sensitivity to the mission parameters? As an example, consider the Kepler mission. The PI has identified a need to determine the frequency of terrestrial planets in the habitable zones of solar-type stars. The quality of the science produced, that is, the number of appropriate planets found per star observed, is some function of a few key mission parameters: instrument precision, detection signal-to-noise ratio, number of stars observed, number of years observed. To determine the sensitivity of the Kepler science goodness, we can establish some merit functions that allow us to study crucial partial derivatives:

$$\frac{\partial(S)}{\partial(\mathrm{SNR})}, \qquad \frac{\partial(S)}{\partial(\mathrm{CDPP})}, \qquad \frac{\partial(S)}{\partial(N_{\mathrm{stars}})}, \qquad \frac{\partial(S)}{\partial(N_{\mathrm{years}})}$$

where science goodness $S = f$(CDPP, SNR, N_{stars}, N_{years}, that is, combined differential photometric precision (CDPP), detection signal-to-noise ratio (SNR), number of stars observed N_{stars}, and number of years observed N_{years}).

Again, the goal is to deliver a system that is robust to variations in the key mission parameters by remaining in the flat portion of the performance curve. Figure 6 provides another, hypothetical example of a merit function in which the sensitivity of the science goodness with respect to instrument precision is plotted. The X marks the intersection of the baseline science need with the required precision. (The former should have driven the latter during synthesis.) The small box around the X represents the region of nominal operation. The region of robust operation is defined by a combination of the science floor (minimum mission success criteria) and the edge of the cliff, which marks the boundary of the region of nongraceful degradation. The mission should strive to work properly within the region of robust operation. Note that the science floor is the most important driver when setting the region of robust operation. In this idealized example, the design has been cost optimized such that the science floor intersects the knee in the curve. (It is of course acceptable and preferable to have some distance between the two.)

Risk Analyses

The need for system-level risk analysis as part of the project validation program was highlighted in the findings of the 1998 MPL Mishap Investigation Board:

> A fault-tree analysis (FTA) was conducted by the project before launch for specific mechanisms and deployment systems where redundancy was not practical. No system-level FTA was formally conducted or documented. . .. The greatest value of system-level FTA is to identify, from a top-down perspective, critical areas where redundancy (physical or functional) or additional fault protection is warranted.[8]

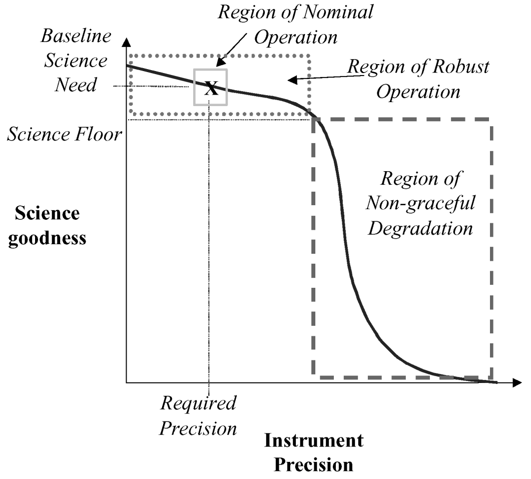

Fig. 6 Science merit function (sensitivity analysis).

(Italics are mine.) Likewise, the MCO mishap investigation noted that a key contributor to that mission loss was “Absence of a process, such as fault tree analysis, for determining ‘what could go wrong’ during the mission.”[9]

Fault tree analysis (FTA) and probabilistic risk analysis (PRA) are useful tools for assessing the robustness of a system to failure modes. An example of a system-level fault tree [an excerpt from the Mars Exploration Rover (MER) terminal descent sequence] is shown in Fig. 7. An FTA is simply an exercise in which the system engineer considers what has to happen for the mission to succeed (or conversely, fail) and then decomposes this in a logical fashion. This is very useful in later phases of system validation, particularly test planning. Also, estimated probabilities of success (or failure) can be assigned to each key component of a fault tree. This is one method of performing a PRA. Engineers often question the accuracy of PRA because it is limited by the assumptions of the analyst on the reliability of very complex processes. However, if one ignores the absolute values of such numbers and instead focuses on the relative differences, PRA can be very useful in highlighting soft spots in the design.

One can then consider the application of selected redundancy or additional fault protection features to augment a soft spot and/or specify appropriate stress tests to prove the system is reliable outside the region of nominal operation. It is fairly standard practice for projects to employ some form of FTA to assess reliability of individual mechanical actuators. Unfortunately, the use of FTA and PRA to study overall system-level robustness is used less consistently. Recent projects such as Mars Odyssey have benefited from the MCO/MPL lessons learned and implemented rigorous FTA and

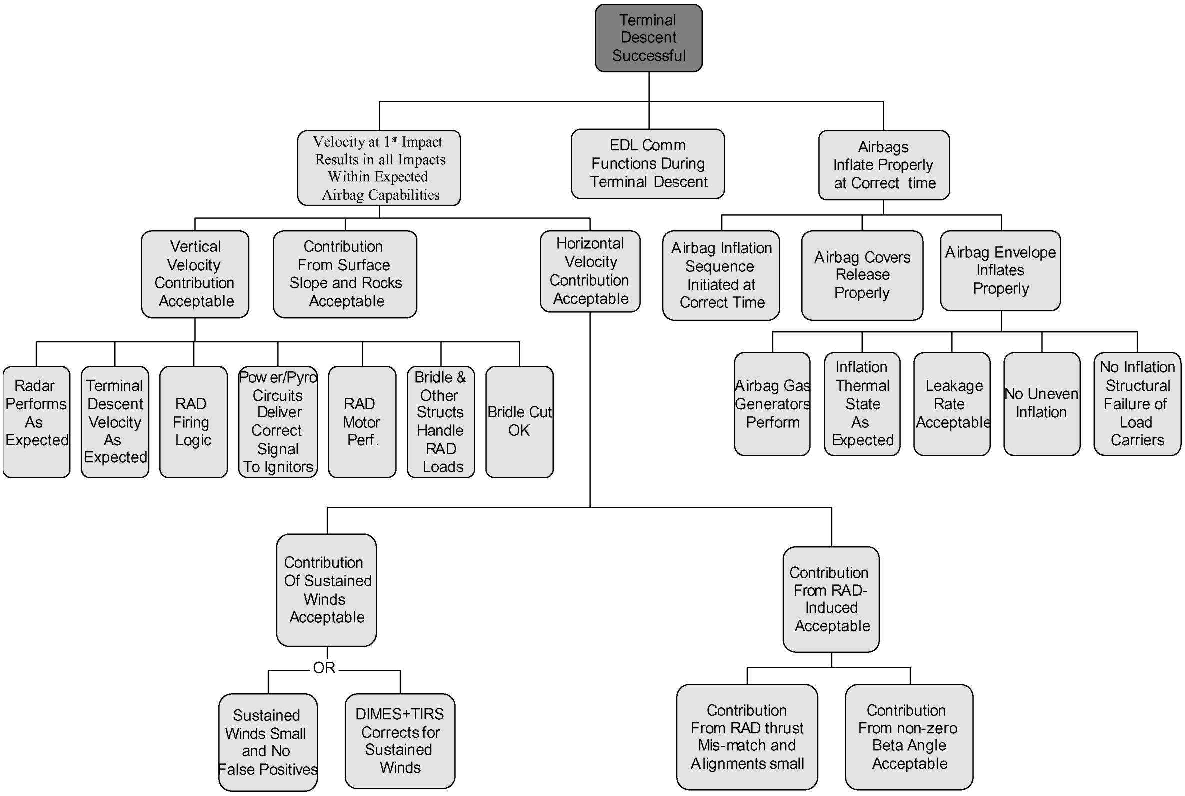
Terminal Descent Successful
Velocity at 1st Impact Results in all Impacts Within Expected Airbag Capabilities
EDL Comm Functions During Terminal Descent
Airbags Inflate Properly at Correct time
Vertical Velocity Contribution Acceptable
Contribution From Surface Slope and Rocks Acceptable
Horizontal Velocity Contribution Acceptable
Airbag Inflation Sequence Initiated at Correct Time
Airbag Covers Release Properly
Airbag Envelope Inflates Properly
Radar Performs As Expected
Terminal Descent Velocity As Expected
RAD Firing Logic
Pow er/Pyro Circuits Deliver Correct Signal To Ignitors
RAD Motor Perf.
Bridle & Other Structs Handle RAD Loads
Bridle Cut OK
Airbag Gas Generators Perform
Inflation Thermal State As Expected
Leakage Rate Acceptable
No Uneven Inflation
No Inflation Structural Failure of Load Carriers
Contribution Of Sustained Winds Acceptable
Contribution From RAD-Induced Acceptable
OR
Sustained Winds Small and No False Positives
DIMES+TIRS Corrects for Sustained Winds
Contribution From RAD thrust Mis-match and Alignments small
Contribution From non-zero Beta Angle Acceptable

Fig. 7 Fault tree for MER terminal descent (excerpt).

risk assessment programs, guidelines for which have been well documented in the literature.[10]

In addition to system-level FTA and PRA, lower-level design risk analyses are necessary to ensure integrated system robustness. A companion to FTA/PRA is the failure modes, effects, and criticality analysis (FMECA). A cognizant engineer for an assembly is responsible for ensuring by analysis that failures in that assembly cannot propagate to other assemblies, for example, a short-circuit condition. FMECA, therefore, tends to be focused on the interfaces between project elements. FMECAs, when combined properly with system FTA/PRA, help ensure the objective of fault containment is met.

As already mentioned, worst-case analysis (WCA) is frequently employed to ensure operation of a particular unit such as an electronic board is robust in the presence of stressing conditions. At even lower levels, individual components are subjected to parts stress analysis (PSA), which are often supplanted by margin tests on the as-built higher-level assembly.

It should be stressed that risk analysis is an activity that spans the project life cycle. It should begin in definition phase, become particularly active around the time of critical design review, and continue into the operations phase and planning for operations (in terms of contingency planning). This "healthy questioning of what could go wrong" is frequently cited as a good feature of a successful project culture.[10] However, for risk analysis to be effective, the person responsible for leading this effort must have sufficient time, broad perspective, and management-supported clout to keep a watchful eye across the project and ferret out risks before they develop into unrecoverable problems. This person is a key link in the project's risk-avoidance chain and having a "Sherlock Holmes approach and a bulldog's disposition to pursue strange indications while the rest of the team is distracted" (well into operations) is vital.[7] One final point about FTA, this activity is ideally done independently, even if it involves project personnel involved in requirements synthesis, taking a fresh look from another angle serves as a form of independent requirements validation.

Early Hardware/Software Testing

When performance sensitivity analyses and design risk analyses identify a potential cliff or soft spot, it is often prudent to perform early tests using engineering model hardware and/or software testbeds to confirm such predictions and/or assess mitigating designs. Again, such tests are only as good as their design. A well-thought-out V&V plan can guide what and how to test during a project's definition phase.

Modeling/Simulation: Roles and Validation

Models (including simulations) play important roles both in requirements validation as well as subsequent verification and system validation. The uses of models in requirements validation were described earlier. For the later phases of V&V, models are often used to bridge the gaps in the system verification and validation effort that cannot be directly tested. Some models (or certain aspects of them) should be considered mission-critical if errors in such areas could mask problems leading to failures in the operations phase of the project. As shown in another finding from the MPL mishap investigation:

> ... the propulsion system, employed analysis as a substitute for test in the verification and validation of total system performance...end-to-end validation of the system through simulation and other analyses was potentially compromised in some areas when the tests employed to develop or validate the constituent models were not of an adequate fidelity level to ensure system robustness.[8]

In a cost-constrained project, the following minimum approach should be used to identify and validate mission-critical models and simulations:

1) In early drafts of the project V&V plan, identify the mission-critical models. (Create verification and system validation storyboards and show where and how models and simulations fit into the overall scheme relative to testing.)

2) Establish requirements on mission-critical model functional capabilities and accuracy.

3) Validate the mission-critical models in terms of those driving requirements.

4) Maintain configuration control of the validated models (treat as mission-critical software).

An example of the V&V storyboard effort is shown in Fig. 8. This addresses the plan for verifying the SIM astrometric performance requirement. Because of practical constraints, this effort involves a mixture of tests and modeling. The critical roles played by some models are circled, such as those needed to propagate the performance of each interferometer (single-baseline) to

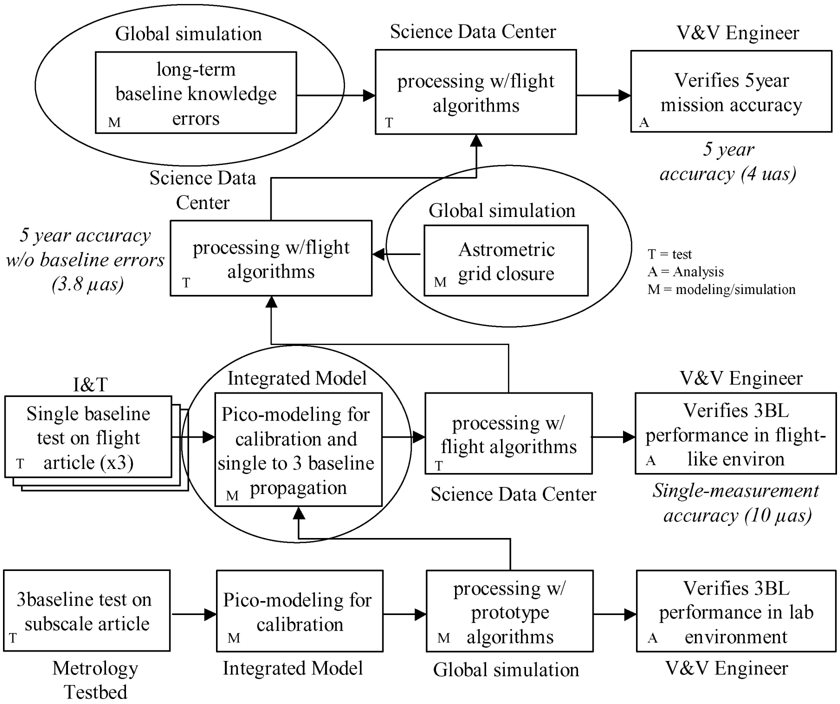
Global simulation
long-term baseline knowledge errors
M
Science Data Center
processing w/flight algorithms
T
V&V Engineer
Verifies 5year mission accuracy
A
5 year accuracy (4 uas)
Science Data Center
5 year accuracy w/o baseline errors (3.8 μas)
processing w/flight algorithms
T
Global simulation
Astrometric grid closure
M
T = test
A = Analysis
M = modeling/simulation
I&T
Single baseline test on flight article (x3)
T
Integrated Model
Pico-modeling for calibration and single to 3 baseline propagation
M
processing w/ flight algorithms
T
Science Data Center
V&V Engineer
Verifies 3BL performance in flight-like environ
A
Single-measurement accuracy (10 μas)
3baseline test on subscale article
T
Metrology Testbed
Pico-modeling for calibration
M
Integrated Model
processing w/ prototype algorithms
M
Global simulation
Verifies 3BL performance in lab environment
A
V&V Engineer

Fig. 8 V&V storyboard for SIM wide-angle astrometric accuracy.

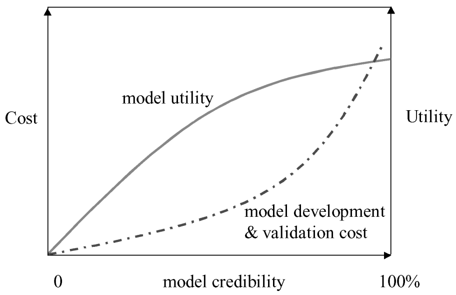

Fig. 9 Model credibility vs cost and utility.

integrated system performance (three baseline) or the incorporation of astrometric grid closure in assessing the ultimate 5-year mission accuracy.

Veteran analysts will admit that model validation is not straightforward in a quantitative, statistical sense and that much emphasis will likely be placed on fairly subjective, qualitative techniques.[11] Using a mix of model validation techniques in complementary fashion can reduce the risk of errors. Such techniques include[12] the following:

1) For face validation, inspect model results by experts on the system being studied to confirm the model seems reasonable and provides the required functional capabilities.

2) For peer review, independently review the theoretical underpinnings and examine model internal components in detail.

3) For the functional decomposition and test, also called piecewise validation, inject test data into individual code modules and compare actual output with predicted output.

4) For comparison or empirical validation, compare performance of the model against performance with the physical system being modeled (or a similar system).

Comparison or empirical validation is a preferred technique and ideally provides quantitative estimates of model credibility or accuracy via cost functions such as Theil's inequality coefficient (TIC) and multivariate statistics (see Refs. 13 and 14).

In practice there are several limitations to this concept, chief among them is that the final form of the system being modeled does not yet exist in the early phases of the project when the model is being used for requirements validation. This situation can often be remedied later in the project when the real system is undergoing testing and can be used to support model evolution, although care must be taken to isolate noise induced by test artifacts from inherent system noise when making comparisons. In the early phase of the project, such model comparisons with real systems can sometimes be accomplished by modifying the model's capabilities to describe a similar existing system (thus, validation by similarity).

Again, project costs can be managed by limiting formal validation efforts to those aspects of models deemed mission critical. Even so, achieving perfect model credibility is typically an unattainable goal within the project cost constraints. The system engineer must establish what level of credibility is sufficient to meet the needs and balance that against model cost and utility requirements. One may decide that project resources are better invested in more comprehensive testing rather than expending additional funds providing an incremental improvement in the credibility and utility of a particular model. The generic relationship between model credibility, utility, and cost is shown in Fig. 9.[14]

Verification and System Validation

Even if requirements and model validation result in a design that should meet the ultimate need, the steps of verification and system validation are required to prove the as-built system in fact does meet those requirements and satisfy the ultimate need. Although the process of verification is well established in the aerospace community, there are two pitfalls that can impact a verification program and must be avoided. First, the use of unvalidated models in covering mission-critical gaps in a testing program is very risky. The quality-assurance concept of using an independent reviewer for verifying a requirement by inspection should apply equally to verification by analysis/modeling. Second, although testing is the most robust method of verification, poorly designed tests can miss problems, provide a false sense of security, and even damage or overstress hardware, resulting in failures during operations. Therefore, it is imperative that test plans be independently reviewed before execution, that tests are conducted by qualified personnel in a buddy-system environment (in which more than one person understands the true intent of the test), and that test reports are critically reviewed by the system engineer to ensure the as-run test did indeed verify compliance with the requirement. The minimum size (lowest cost) of a V&V team is that which ensures there are no reasonable single-point-failures in terms of the integration and test team missing something critical.

Another related concept is that of the incompressible test list, which identifies the minimum set of tests that must be performed before certain project milestones such as launch, regardless of schedule and budget pressures. Once established, any changes to this list must be approved by system engineering and project management.

As already mentioned, validation does not follow the same well-established policies as verification. When starting to put together a system validation matrix, the engineer may struggle with how to start. However, the same techniques discussed earlier in the section on requirements validation can be applied in generating such a matrix, which will define what validation tests are performed.

Validation should include operational readiness tests (ORTs) that assess how the end-to-end system will really perform when all of the flight and ground hardware, software, people, and operational procedures come together. Cross-system compatibility tests or scrimmages to validate things such as flight–ground interfaces are useful precursors and complementary activities to ORTs. Another key component of system validation is stress testing and simulation, in which system robustness to variations in performance and fault conditions are assessed. The region of robust operation in Fig. 6 shows the space over which system performance should be tested. Likewise, the results from system FTA and PRA should guide the definition of mission scenario tests, which include fault injection. The importance for such stress testing and simulation was cited in the WIRE failure report: "Testing only for correct functional behavior should be augmented with significant effort in testing for anomalous behavior."[15] Likewise from the MPL mishap report, "The flight software was not subjected to complete fault-injection testing. System software testing must include stress testing and fault injection in a suitable simulation environment to determine the limits of capability and search for hidden flaws."[8] Some projects rightly claim they cannot do full fault-injection testing with the flight system without incurring excessive risks or costs. However, the solution is to invest in software testbeds to do this. Again, an early V&V plan can identify the needs for such testing and the required testbed capabilities.

Several decades of lessons learned in deep-space mission implementation have been captured in a set of design principles and flight project practices at NASA's Jet Propulsion Laboratory, which in addition to the V&V principles already discussed, address the following:

1) Perform a mission design verification test (MDVT) to validate the end-to-end flight–ground system by covering mission-enabling scenarios, for example, science operations, trajectory correction maneuver, safing, cruise.

2) As a minimum, perform validation of the launch sequence and early flight operations on the flight vehicle using the launch version of flight software.

3) Test as you fly and fly as you test, for example, using flight sequences, flightlike operating conditions, and the same software functionality.

4) Stress testing shall consider single faults that cause multiple-fault symptoms, occurrence of subsequent faults in an already faulted state, etc.

5) Perform system level electrical plugs-out test using the minimum number of test equipment connections.

6) In addition to usual real-time data analysis, comprehensive non-real-time analysis of test data shall be performed before considering an item validated or verified.

7) Testing is the primary, preferred method for design verification.

8) Results of V&V by modeling, simulation, and/or analyses must be independently reviewed.

9) Changes made to the system to address issues found during testing must be reverified by regression testing.

10) Verification by visual inspection, particularly for mechanical clearances and margins, for example, potential reduced clearances after blanket expansion in vacuum, must be performed on the final as-built hardware before and after environmental tests.

11) Verification of all deployable or movable appendages and mechanisms must include full-range articulation.

12) The navigation design must be validated by peer review by independent subject matter experts.

13) Mission operations capabilities, for example, flight sequences, command/telemetry databases, displays, must be employed in system testing of the flight system.

V&V Checklist

In summary, by combining the principles outlined in this paper, system engineers can use the following checklist to guide V&V efforts.

I. Planning:

1) Draft V&V plan in early phase B and determine risk vs cost posture.

2) Use risk ratings to identify requirements that need formal validation.

3) Consider V&V as a three-dimensional process (width, depth, and time).

4) Assign a senior system engineer to risk analysis/V&V who has the time, perspective, and clout to keep a watchful eye project wide on potential problems (extending into operations).

II. Requirements validation techniques:

1) Requirements validation must address (with an emphasis on get it right the first time): completeness, correctness, achievability, verifiability, and robustness.

2) End-to-end functional flow diagrams.

3) Error budgets/models (top-down and bottom-up):

a) Beware of systematic errors.

b) Understand error reduction and calibration residuals.

4) Performance sensitivity analyses: identify the robust region of operation, merit functions (amount and quality of products accomplished by mission), and stressing/envelope analyses (Monte Carlo simulation).

5) Risk analyses as an independent check on requirements and design:

a) System FTA. (What must work?)

b) PRA. (Where are the soft spots?)

c) FMECA. (What are the failure modes and are faults contained?)

d) WCA. (How do things respond to stressing conditions?)

6) Early hardware/software testing.

III. Model validation techniques:

1) Story boards to identify mission-critical models (those bridging gaps in test program).

2) Use complementary techniques to validate formally mission-critical models: face validation, peer review, functional decomposition and test, and comparison/empirical validation.

IV. System V&V techniques:

1) Test as you fly, fly as you test.

2) Validate system performance and functionality across both the nominal and robust regions of operation.

3) Use FTA results to establish V&V requirements and matrices.

4) Assess system robustness with mission scenario tests.

5) Assess interaction of hardware, software, people, and procedures with ORT and cross-system scrimmages.

6) Preferred V&V method: testing.

7) Suitable alternatives to test are to

a) span gaps in test regime by analysis with validated models/simulations

b) independently review of any results from such modeling/analysis

c) use software testbeds to augment fault injection testing

Conclusions

It is certainly possible to implement relatively low-cost, fast-paced missions without taking excessive risks, but doing so requires that projects commit to following some form of rigorous V&V program. Although there is no one size fits all V&V program, projects should follow some minimum set of guiding principles such as those described in this paper when selecting an appropriate cost–risk balance.

Acknowledgments

The work described here was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract to NASA. The author is grateful for useful discussion with colleagues including Gentry Lee, G. Mark Brown, Chet Sasaki, Christian Liebe, Ross Jones, and Neil Yarnell at the Jet Propulsion Laboratory (JPL) and Bill Borucki at NASA Ames Research Center. The Mars Exploration Rover terminal descent fault tree was provided by Robert Mitcheltree of JPL and NASA Langley Research Center. Mike Margulis of Lockheed Martin Missiles and Space Systems and JPL's Peter Kahn provided input on the Space Interferometry Mission astrometric performance validation flow. Feedback from the reviewers was also useful in expanding the content of this paper.

References

[1] Kerr, R. A., "NASA's New Road to Faster, Cheaper, Better Exploration," *Science*, Vol. 298, Nov. 2002, pp. 1320–1322.

[2] Ansorge, W. R., "Assembly, Integration, Verification and Validation in Extremely Large Telescope Projects, a Core Systems Engineering Task," *Telescope Structures, Enclosures, Controls, Assembly/Integration/Validation, and Commissioning*, edited by T. A. Sebring and T. Andersen, Proceedings of the SPIE, Vol. 4004, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, 2000, pp. 559–567.

[3] Grady, J. O., *System Validation & Verification*, CRC Press, Boca Raton, FL, 1998, pp. 74, 75.

[4] Hoyle, D., *ISO9000 Quality Systems Handbook*, 3rd ed., Butterworth–Heinemann, 1998, p. 234.

[5] Hoban, F., and Hoffman, E., *NASA Systems Engineering Handbook*, NASA SP-610S, June 1995, Chap. 2.5.

[6] Rosenberg, L. H., "Verification and Validation Implementation at NASA," *Crosstalk*, Vol. 14, No. 5, 2001, pp. 12–15.

[7] Euler, E., Jolly, S., and Curtis, H. H., "The Failures of the Mars Climate Orbiter and Mars Polar Lander: A Perspective From the People Involved," *Advances in the Aeronautical Sciences*, Vol. 107, 2001, pp. 635–656.

[8] Casani, J., "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions," URL: http://spaceflight.nasa.gov/spacenews/releases/2000/mpl/mpl_report_1.pdf, 22 March 2000 [cited 8 Aug. 2003].

[9] Stephenson, A. G., "Report on Project Management Within NASA by the Mars Climate Orbiter Mishap Investigation Board," URL: http://www.space.com/media/mco_report.pdf, 13 March 2000 [cited 8 Aug. 2003].

[10] Beutelschies, G., "That One's Gotta Work—Mars Odyssey's use of a Fault Tree Driven Risk Assessment Process," *Proceedings of the IEEE Aerospace Conference*, Paper 6599-2, IEEE Publications, Piscataway, NJ, 2001.

[11] Arthur, J. D., Sargent, R. G., Dabney, J. B., Law, A. M., and Morrison, J. D., "Verification and Validation: What Impact Should Project Size and Complexity have on Attendant V&V Activities and Supporting Infrastructure," *Proceedings of the IEEE: 1999 Winter Simulation Conference*, edited by P. A. Farrington, Inst. of Electrical and Electronic Engineers, New York, 1999, pp. 148–155.

[12] Youngblood, S. M., and Pace, D. K., "An Overview of Model and Simulation Verification, Validation, and Accreditation," *Johns Hopkins APL Technical Digest*, Vol. 16, No. 2, 1995, pp. 197–206.

[13] Smith, M. I., Hickman, D., and Murray-Smith, D.J., "Test, Verification, and Validation Issues in Modelling a Generic Electro-Optic System," Infrared Technology and Applications XXIV, Proceedings of the SPIE, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, Vol. 3436, 1998, pp. 903–914.

[14] Balci, O., "Principles of Simulation, Model Validation, Verification, and Testing," *Transactions of the Society for Computer Simulation International*, Vol. 14, No. 1, 1997, pp. 3–12,

[15] Branscome, D. R., "WIRE Mishap Investigation Board Report," URL: http://klabs.org/richcontent/Reports/wiremishap.htm, 8 June 1999 [cited 3 Aug. 2003].

J. Martin
Associate Editor