Addressing the hard factors for Command File Errors by Probabilistic Reasoning

Leila Meshkat & Larry Bryant

Jet Propulsion Laboratory

California Institute of Technology

Abstract

Command File Errors (CFE) are managed using standard risk management approaches at the Jet Propulsion Laboratory. Over the last few years, more emphasis has been made on the collection, organization, and analysis of these errors for the purpose of reducing the CFE rates. More recently, probabilistic modeling techniques have been used for more in depth analysis of the perceived error rates of the DAWN mission and for managing the soft factors in the upcoming phases of the mission. We broadly classify the factors that can lead to CFE's as soft factors, which relate to the cognition of the operators and hard factors which relate to the Mission System which is composed of the hardware, software and procedures used for the generation, verification & validation and execution of commands. The focus of this paper is to use probabilistic models that represent multiple missions at JPL to determine the root cause and sensitivities of the various components of the mission system and develop recommendations and techniques for addressing them. The customization of these multi-mission models to a sample interplanetary spacecraft is done for this purpose.

Introduction

Command File Errors (CFE's) occur when (a) there is an error in a command file that was sent to the spacecraft, (2) an error in the approval, processing or unlinking of a command file that was sent to the spacecraft or (3) an omission of a command file that should have been sent to the spacecraft[2,8]. We broadly categorize the root causes for a CFE as soft factors or hard factors[1]. Soft factors are those that mainly contribute to the operator cognition, for instance the level of training, situational awareness or management and organizational factors. Hard factors are those factors that have tangible products associated with them, such as software, hardware and operations process procedures and documentation. The hardware and software includes those used for simulating commands and verification and validation of spacecraft. The background section below describes the processes involved in creating these hard factors. The approach section describes the study conducted here in collecting and analyzing the command file errors associated with the hard factors as well as the insight provided by this study.

Background

The investigation into the root causes of command file errors has led us to abstract these causes and represent them in compact, executable models.

Figure 1: Abstract model

A simple abstraction of the model is shown in figure 1. This figure shows that command file errors can be caused due to slips or mistakes. Mistakes occur when the operators' decision in terms of the course of action is flawed and slips occur when the execution of the decision is incorrect. In order to manage CFE's, it's important to take into consideration all their possible causes, and the inter-relationships between these causes and each of their effects. This can be represented with a Directed Acyclic Graph (DAG). Furthermore, the probabilistic relationships between the nodes in this network can be captured and analyzed using Bayesian techniques [3,4]. Hence the applicability of Bayesian Belief Networks (BBNs) for the systems analysis and management of CFEs.

Deliberation on the soft factors that result in CFE's using BBN's led us to quantitatively establish the influence of management and organizational factors, which, in turn affect the situational awareness and inter and intra team communications of flight system operations teams [10,11]. Figure 2 elaborates on the hard factors that have been associated with command file errors. These are classified as the simulators, which include hardware simulations and hardware with software in the loop simulators, the actual software on the spacecraft, which includes the ground and flight software and the process procedures and documentation of the mission system.

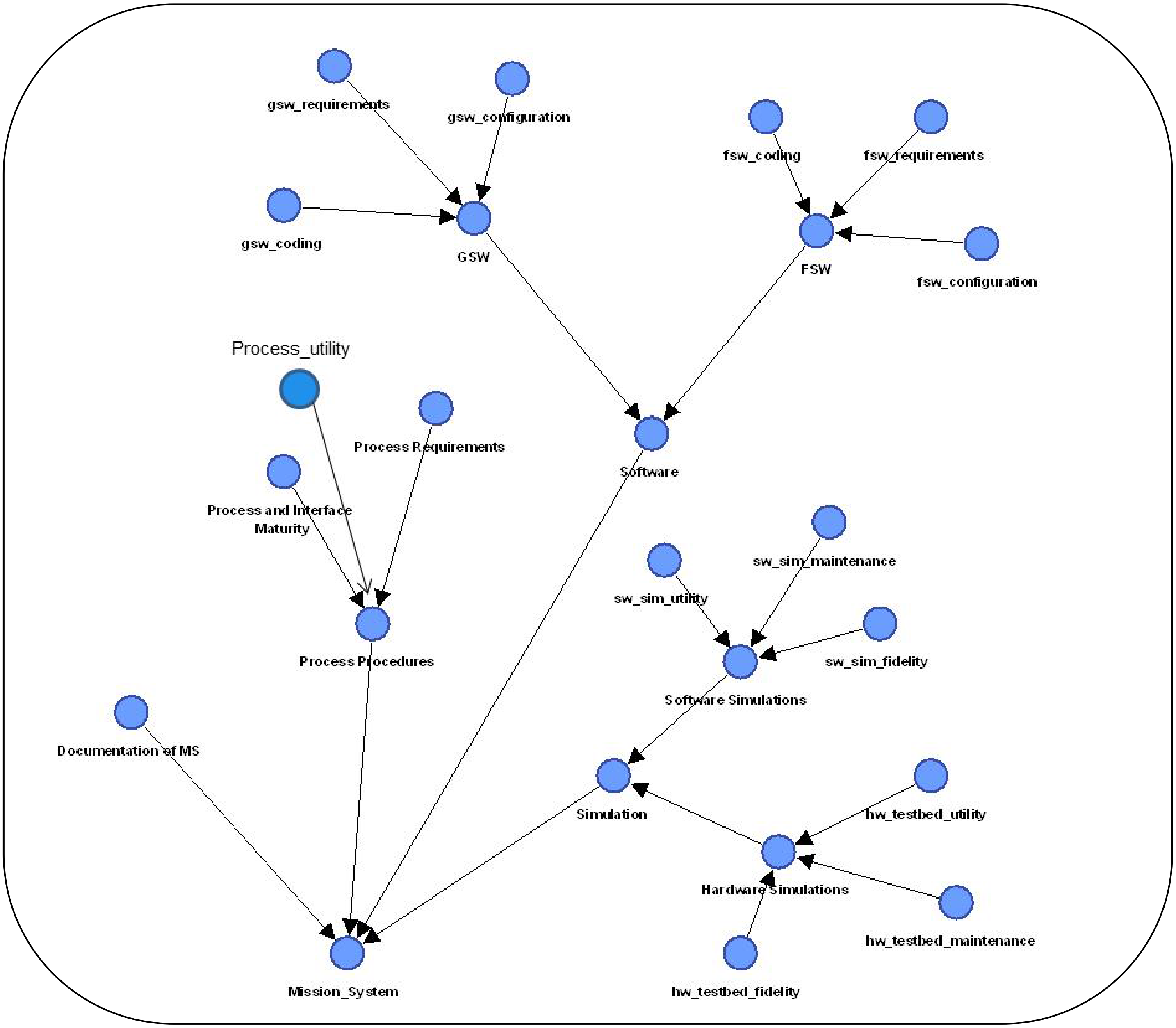

Figure 2: Hard Factors associated with Command File Errors

Errors that are occur due to process procedures have been traced back to inadequacies in the requirements, the process and interface maturity and the utility of the procedures. Errors associated with ground and flight software are traced back to the coding, requirements or configurations associated with them. As far as the simulations are concerned, their errors are traced to the utility, fidelity or maintenance of the simulators, whether software only or hardware based simulators with software in the loop.

Hard Factors:

A study across sixteen JPL missions and eleven years indicates that roughly one third of command file errors are attributed to hard factors. More in-depth examination of a specific flight project indicates that even though the ones attributed only to hard factors are only one-third of the whole, more than half of all command file errors have both hard and soft factors associated with them. A simple example is when a flight rule is defined somewhat ambiguously in the procedures and its effect is also not communicated across the teams. So the procedure part of this is a hard factor issue and the communication part a soft factor. And hence the root cause is a combination of hard and soft factors.

In this section, we discuss each of the main categories of hard factors separately and explain the current state of practice in terms of how they are developed and maintained throughout the lifecycle of a flight project.

Simulations

Modeling & Simulation techniques are widely used across projects throughout their lifecycles. Although each flight project at JPL does have its own simulations, there are also a suite of tools and planning and sequencing products that are provided by a multi-mission institute, called the MGSS (Multi-mission Ground Systems & Services) that projects then tailor to their own needs.

Typically, the sophistication of the simulation tools that are used for command generation depends on the budget of the flight project in question. In particular, the hardware simulations, which are called Systems Test Laboratories (STLs) may be very complete and provide an accurate engineering model of the spacecraft. They may have their own command and data handling system, recorders (tape or digital) and a power batteries. There may even be engineering models of instruments which are fully functional and breadboards where they test commands. The flight software which is loaded on to the STL's is typically the same flight software that is on the spacecraft. So the interaction between the commands and the flight software on the spacecraft is simulated from end to end before sending the commands to the actual spacecraft. The various software tools and test-bed elements are often developed and used first during the ATLO (Assembly, Test & Launch Operations) phase and carried through to the Operations phase.

As the budget becomes more constrained, the level of fidelity of the hardware engineering models are typically reduced. For instance, there may be a block of wood that simulates the weight of the instruments instead of real instruments. There may also be a power battery simulator instead of replicas of the batteries and solar cells. Some parts of the system may also be represented with their software models that provide the inputs and outputs and hence provide continuity (for instance in the case of various thermal modes this may be an alternative to having a full thermal system engineering model).

We have broadly classified simulations as hardware based and software based. And elements that may lead to their inadequacy are their lack of fidelity, the lack of maintenance such that they don't reflect the latest state of the spacecraft or the lack of user-friendliness such that it's too hard to use them or interpret their results. Software based simulations simply simulate the end to end spacecraft system rather than providing an actual physical model of it.

Software

The ground software is software which is used to generate commands. It often includes a plethora of tools which are initially obtained from the MGSS and then customized for the project at hand. Clearly, the inadequacy of the software used for generating commands could easily contribute to errors in the command files. The root cause of this inadequacy could be either in the requirements associated with the ground software, the configuration of the software and how the various tools work in concert with each other or the actual coding of the software.

Commands that are sent to the spacecraft interact with the flight software and it's this interaction that may result in an error. Therefore, the flight software(fsw) may also contribute to CFE's. Often these errors occur due to the inadequacy of capturing the interactions between the command and the fsw in the requirements documents for the fsw. The configuration and coding of this software may also contribute to such errors.

Procedures

The procedures documents and flight rules aim to provide a guideline for the development of command files. They therefore must contain sufficient and accurate information about both the flight and ground system and the process for creating commands for every phase of the flight project.

A significant portion of CFE's that occur due to the hard factors can be attributed to procedures. On one hand, there is often a lack of maturity in the procedures at the offset of the project and only when the spacecraft has been operated on for a while do they gain the necessary maturity. On the other hand, sometimes the procedures are accurate but are not followed because they are too hard and complex to understand. And of course sometimes the requirements based on which the procedures are created are not complete, and specifically, don't address the interaction effects between the various elements involved in the generation of command files.

Flight System Documentation

The flight system is sometimes built by contractors but managed and operated at JPL. In these instances, the quality of the documenation associated with the flight system is very significant when it comes to operating the system.

Approach

The approach used for this study is to (1) determine the frequency of errors associated with each of the nodes for a sample flight project (2) use this data, combined with subject matter expertise to populate an associated BBN. This BBN will then be assessed and analyzed to provide insight into the sensitivity of command file errors to the hard factors and how projects might optimize their investments in terms of reducing the CFE rates. It's important to note that the models in this paper are built based on a combination of expert opinions and subjective data. These models will be iterated upon with the experts and customized to individual projects several times before they are infused within the project team and used to provide insight.

Complete Model

In this section, we provide context by describing quantitative experiments that are based on the full BBN model representing the causes of the CFE's. The next section then focuses on conducting probabilistic reasoning for root cause analysis of the hard factors per se.

Figure 3 shows a screen shot of the actual BBN model in the SAMIM [ref] tool query mode. The circles indicate the nodes of the BBN and the rectangles the probabilities associated with them. This model has been generated by combining data elicited from experts with actual data from the problem failure reporting database for a sample inter-planetary spacecraft.

	Average	Early Phase or Critical Event	Cruise Phase
Command Error	1.65%	1.84%	0.78%
Mistake	23.57%	26.24%	11.15%
Slip	23.50%	26.16%	11.11%
Operator Cognition Incorrect	21.90%	24.24%	9.68%
Mission System Inadequate	2.21%	2.73%	1.66%
Situational Awareness Low	22%	25.81%	16.85%
Skill Level Low	7%	7.00%	7.00%
Operational Complexity High	50%	65.00%	35%
Process Procedures Inadequate	2.50%	3.00%	1.50%
Software Inadequate	1.75%	2.15%	1.45%
GSW Inadequate	3%	7.00%	4%
FSW Inadequate	3.75%	3.75%	3.25%
Simulation Inadequate	2.77%	3.71%	1.54%
Hardware Simulation Inadequate	6.85%	8.55%	4.72%
Software Simulation Inadequate	7.00%	10.00%	3%

Table 1: Comparison of probability of key nodes during different mission phases.

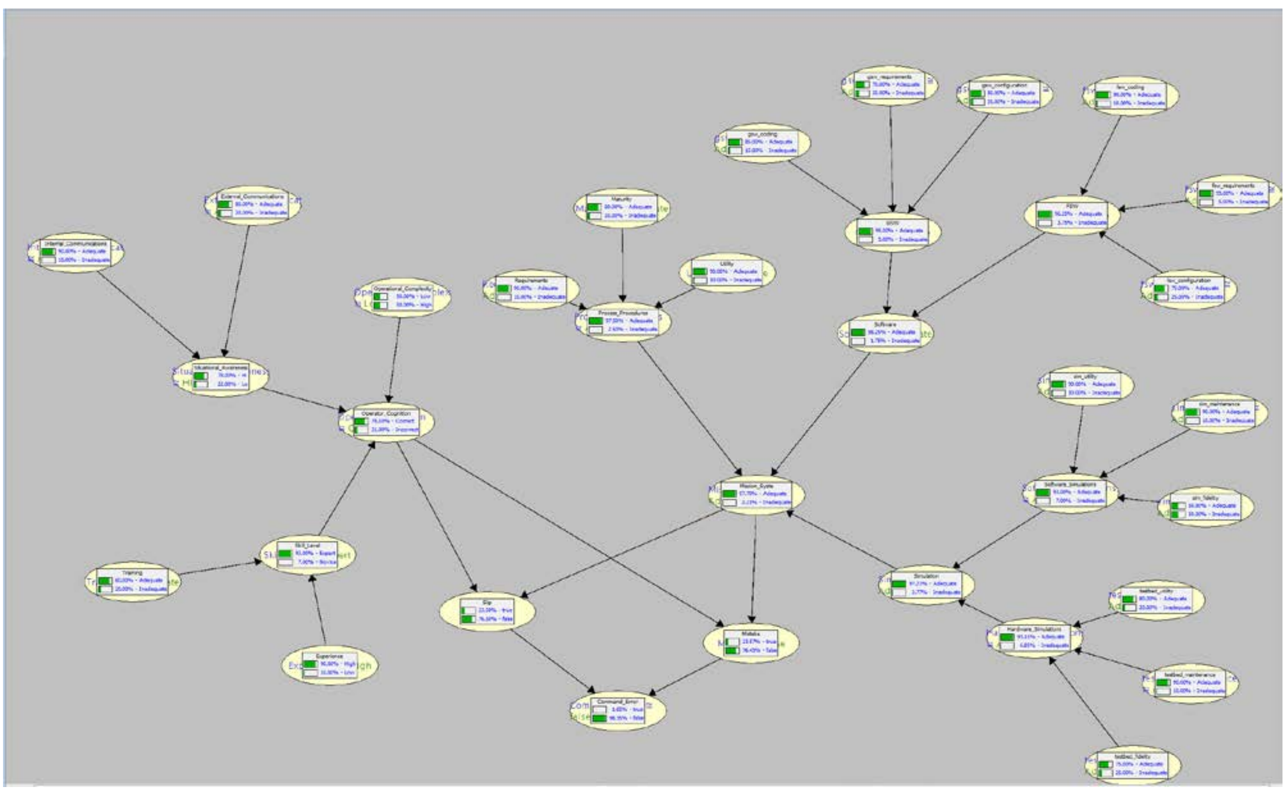

Figure 3: Screen-shot of the BBN model in the SAMIAM tool Query mode – with posterior probabilities.

Table 1 provides a comparison of the probability of key nodes in the BBN during the various mission phases. A graph corresponding with this table is shown in figure 4.

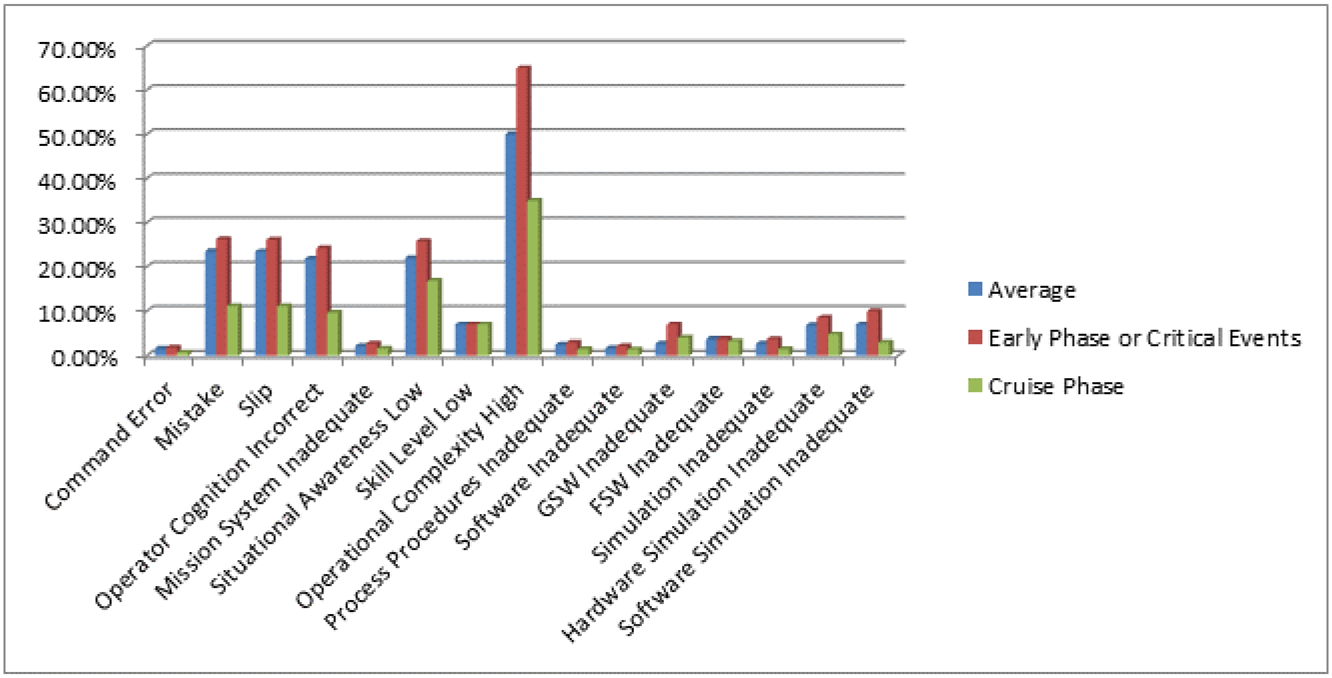

Figure 4: Graph corresponding with the probability of key nodes shown in table 1.

As it can be seen in table 1 and figure 4, the key drivers for the difference in the command file error rates during the various phases are the increase in the operational complexity of the mission which in turn affects the cognition of the operators. Furthermore, during the critical phases of the mission, since we are dealing with a lot of uncertainty, the process procedures associated with commanding are less mature and the fidelity of the software and hardware simulation models lower. Another key difference is in the situational awareness which in turn depends on the communications between teams as well as the simulation models available.

Hard Factors

In this section, we focus on the hard factors, or the Mission Systems branch of the BBN and conduct some experiments to determine (1) the sensitivity of the adequacy of the Mission System, as it relates to command file errors to each of the elements of the simulation, software and process procedures and (2) areas where we could get the biggest return by providing improvements.

The sensitivity of the Inadequacy of the Mission system to the adequacy of each of the elements that contribute to the hard factors is shown in table 2. These numbers are obtained by assessing the level of inadequacy of the mission system for the two extreme cases of each variable: when it is 100% adequate and when it is 100% inadequate. Note that the sensitivity to Simulation in general is much higher than the sensitivity to each of the elements of simulation (hardware simulations & software simulations). This indicates a significant interaction effect. Simulations are used concurrently and the inadequacy of one type may be mitigated by the adequacy of another. Same statement holds for the process procedures and software factors.

			Sensitivity of Mission System Inadequacy to:
Simulation			14
	Hardware Simulations		2.8
		utility	0.48
		maintenance	0.26
		fidelity	0.25
	Software Simulations		2.8
		utility	0.28
		maintenance	0.28
		fidelity	0.28
Software			15
	Ground Software		3
		coding	0.3
		requirements	0.15
		configuration	0.3
	Flight Software		3
		coding	0.3
		requirements	0.15
		configuration	0.3
Process Procedures			15
		utility	1.5
		requirements	0.75
		maturity	0.75

Table 2: Sensitivity Analysis for the Hard Factors

In order to determine where we would be able to invest to get the biggest improvement, we need to take into consideration (1) the current value of the variable in question as opposed to its worst case scenario and (2) examine the root level nodes which are actionable. Table 3 shows the Improvement Potential of each of these root level nodes. This is obtained by finding the difference in the level of adequacy of the mission system that results by changing the level of adequacy of the variable from its current value to a state in which it is 100% adequate. A graph corresponding to this table is seen in figure 4. It's clear that the area with the biggest improvement potential is the utility of the procedures. The next area seems to be the configuration of the flight software and the utility and fidelity of the hardware simulators.

Root Level Node	Improvement Potential
hw-sim-utility	0.05
hw-sim-maintenance	0.02
hw-sim-fidelity	0.05
sw-sim-utility	0.03
sw-sim-maintenance	0.03
sw-sim-fidelity	0.03
gsw-coding	0.04
gsw-requirements	0.04
gsw-configuration	0.03
fsw-coding	0.03
fsw-requirements	0
fsw-configuration	0.06
proc-utility	0.15
proc-requirements	0.03
proc-maturity	0.03

Table 3: Improvement Potential for Root Nodes

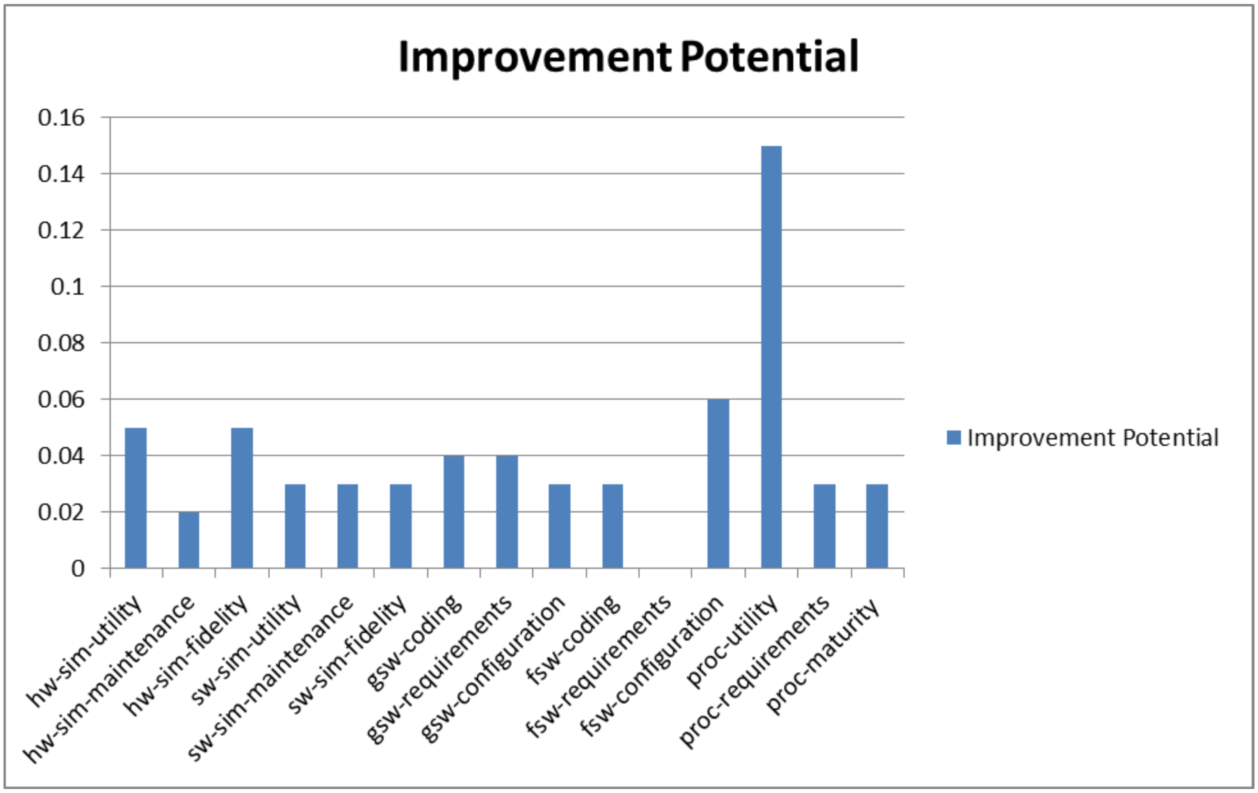

Figure 4: Graph Representing the Improvement Potential for Root Nodes.

Summary & Future Directions

We classified the factors that can lead to command file errors into hard and soft factors. Hard factors are those elements of the system which are tangible, such as the simulation test-beds, the ground and flight software and the process procedures. We then focused on the hard factors and discussed how they are developed and what causes them to contribute to command file errors. A Bayesian Belief Network which represents the probabilistic and causal relationships within the system used to generate command files was then presented and some experiments were done to determine the sensitivity of the Mission System to each of the elements of the hard factors and the areas with the most improvement potential which could yield the highest return on investments.

We are currently conducting empirical studies on a set of flight projects in order to validate the various findings of the BBN models. These studies include regression analysis, design of experiments and factor analysis.

In the future, we plan on iterating on the model used for this study with the subject matter experts on the flight project in question several times until the results seem completely agreeable to them and then infuse it within the teams for assistance in making decisions for managing the risk of command file errors for the upcoming critical activities of the mission. Further, we will be summarizing results of studies on a number of flight projects to provide insight to the MGSS to help with improving their current multi-mission infrastructure.

Acknowledgements

The work reported in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with NASA. The authors would like to acknowledge the material provided by Grant Faris as well as his invaluable insights and advice. We would also like to acknowledge Bruce Wagoner for the formation of the structure of the model and Chuck Scott and Ed Hirst for providing us with expert data.

References

1. L. Meshkat & L. Bryant, “Managing the Risk of Command File Errors”, AIAA Aerospace Sciences Meeting, January 2013.
2. Grant Faris, “Proposed Institutional Command File Error Definition, Categories, and Metrics". 3 April 2008.
3. P. Trucco, E. Cagno & O. Grande, “ A Bayesian Belief Network Approach for Integrating Human and Organizational Factors in Risk Analysis: A Case Study for the Maritime Industry.

4. L. Meshkat, J. B. Dugan, G. Faris, "Command Error Modeling", International Symposium for Software Reliability Engineering, November 2011
5. L. Meshkat, "A Systems Modeling Approach for Risk Analysis for risk management of command file errors." , Space Operations Workshop at JPL, April 2012
6. G. B. Faris, L. W. Bryant "Improving Operations: Metrics to Results".
7. Kelly Bezjak and Bruce Waggoner, "Mission Operations Flight School", 2010.
8. H. Kwong-Fu, C. Scott, R. Smith, G. Chin, " Command File Error Metrics Working Group Analysis & Results", 2008.
9. L. Meshkat, L. Bryant, & B. Wagoner " Modeling to Improve the Risk Reduction Process for Command File Errors", AIAA Space Conference, September 2013.
10. L. Meshkat, L. Bryant, B. Wagoner & R. Thomas, "Soft Factors & Space Mission Failures: Quantifying the Effect of Management Decisions", AIAA Science & Technology Forum, January 2014.
11. The Automated Reasoning Group at UCLA "Sensitivity Analysis, Modeling, Inference & More" http://reasoning.cs.ucla.edu/samiam/"