Requirements Based System Level Risk Modeling

Leila Meshkat, Ph.D., Jet Propulsion Laboratory, California Institute of Technology, CA.
Steven Cornford, Ph.D., Jet Propulsion Laboratory, California Institute of Technology, CA
Martin Feather, Ph.D., Jet Propulsion Laboratory, California Institute of Technology, CA.

Key Words: risk, requirements, relational dependencies, sequential dependencies.

SUMMARY & CONCLUSIONS

Space exploration missions are often characterized by multiple phases and each phase in turn satisfies some objective or requirement. The success of the mission is measured by the degree to which these requirements are satisfied. Missions either aim to demonstrate a new technology, or to obtain new science data or a combination of both of these. During the mission design process, numerous trade studies are conducted between cost, performance and risk. At a very high level, the goal is to maximize the probability of achieving the most science return (or demonstrating the most technology) at the least possible cost. We consider the problem of maximizing this probability by quantifying the degree of importance of each requirement and it's probability of being satisfied. The probability of a requirement being satisfied, in turn, is assessed by finding the aggregate of the probability of all the possible events that could prevent it from being satisfied.

We assume a complete list of the requirements, the relevant risk elements and their probability of occurrence and the quantified effect of the risk elements on the requirements. In order to assess the degree to which each requirement is satisfied, we need to determine the effect of the various risk elements on the requirement. The complexity arises due to the fact that various risk elements that effect a requirement in question are not necessarily independent. Moreover, in order to compute the weighted average of the requirements, it's important to take into consideration their dependencies. Therefore we carefully define the relationships between the elements within each category (intra-category) and the elements between the two different categories of risk and requirements (inter-category).

1. INTRODUCTION

Assessment of the ability of a system (mission) to perform its required functions in a given time frame can be accomplished using appropriate modeling techniques. One such technique is a quantitative representation of the dependencies between the system (mission) requirements and their respective risk elements. The success criteria for the system are identified by the set of requirements that it is intended to achieve. Requirements have different weights from a systems perspective; some may be more important than others. The degree of system success is expressed as a weighted sum of the requirements attainment. Any event that prevents one or more requirements from being satisfied is a risk element. Risk elements appear in various forms. In some cases, they are decomposed into more atomic events. In other cases, they are combined to yield further risk elements. Both the combined and the decomposed elements may in turn effect one or more of the requirements specified in the system or mission success criteria. Moreover, requirements could be subject to the same process. They could be decomposed into more atomic elements, or combined to yield higher-level requirements.

The problem that we address in this paper is assessing the expected degree of success of the system or mission based on the degree to which each requirement is satisfied and the relative weight of the requirements. We assume a complete list of the requirements, the relevant risk elements and their probability of occurrence and the quantified effect of the risk elements on the requirements. In order to assess the degree to which each requirement is satisfied, we need to determine the effect of the various risk elements on the requirement. The complexity arises due to the fact that various risk elements that effect a requirement in question are not necessarily independent. Moreover, in order to compute the weighted average of the requirements, it's important to take into consideration their dependencies. Not taking these dependencies into account results in double counting some elements and hence an incorrect measure of the success of the system or mission. Therefore, the challenge that we encounter is carefully defining the relationships between the elements within each category (intra-category) and the elements between the two different categories of risk and requirements (inter-category). In addition, appropriately quantifying and modeling inter and intra category dependencies are crucial in estimating the exact measures of system success.

We consider the degree to which each requirement is satisfied to be a function of the risk elements that affect it. The probability of occurrence of higher-level risk elements are assessed from their basic events using fault tree analysis techniques.

2. MOTIVATION

Space exploration missions are often characterized by multiple phases and each phase in turn satisfies some objective or requirement. The success of the mission is

measured by the degree to which these requirements are satisfied. Missions either aim to demonstrate a new technology, or to obtain new science data or a combination of both of these. During the mission design process, numerous trade studies are conducted between cost, performance and risk. At a very high level, the goal is to maximize the probability of achieving the most science return (or demonstrating the most technology) at the least possible cost. We consider the problem of maximizing this probability by quantifying the degree of importance of each requirement and it's probability of being satisfied. The probability of a requirement being satisfied, in turn, is assessed by finding the aggregate of probability of all the possible events that could prevent it from being satisfied.

At JPL and NASA, the Defect Detection and Prevention (DDP) tool has been created by Cornford et al. [1,4,5] to address this optimization problem. In the following section, we briefly describe this tool.

2.1 Defect Detection and Prevention

"Defect Detection and Prevention" (DDP), is a simple risk model designed for application early in the lifecycle, when information is sparse yet the capability to influence the course of the development to follow is large. Cornford originally conceived of DDP specifically to facilitate assurance planning [8]. The core idea of DDP is to relate three sets of information:

1. "Requirements" (what you want to achieve).
2. "Risk Elements" (what can get in the way of attaining those objectives).
3. "Investments" (what you can choose to do to overcome the problems).[1]

In DDP, relationships between these items are *quantitative* (e.g., *how much* a Risk Element, should it occur, detracts from an Requirement's attainment). Such a quantitative treatment is key to DDP's realization of the vision of "risk as a resource", as espoused in [7]. This is one of the key ways that DDP differs from many of the purely qualitative approaches (e.g., QFD [9]) usually employed early in the life cycle.

Cornford's initial experiments used Microsoft Excel® spreadsheets to manually explore the utility of the process. Positive results then led to development of custom software for the DDP process [1]. Supported by this software, DDP has been applied to assess the viability of, and planning for, the development of novel technologies and systems for use on space missions [6,7].

The core steps of a DDP risk study are:

1. Represent the success requirements of the spacecraft mission as DDP's "Requirements". User-provided weightings indicate the relative importance of these.
2. Represent the plethora of all kinds of risks that could impede attaining those objectives as DDP's "Risk Elements". These can encompass a wide range of concerns: programmatic, technical, infrastructure, management and resources.
3. Capture the extent to which each Risk Element, should it occur, would detract from attainment of each Requirement. These become DDP's quantitative "impact" links. Note that multiple Risk Elements, to varying degrees, can impact a Requirement, and similarly a Risk Element can impact multiple Requirements.
4. Represent the options for reducing risk, including preventative measures, development-time tests and analyses (which, by revealing the presence of problems, allow for their correction prior to flight), as DDP's "Investments". Each of these has associated resource costs (e.g., dollars, time, map, power). Investments may include technology investments, design/architectural options, tests, analyses, process controls, and operational solutions.
5. Capture the extent to which each Investment, should it be applied, would reduce each Risk Element. These become DDP's quantitative "effect" links. Note that multiple Investments, to varying degrees, can affect a Risk Element, and similarly an Investment can impact multiple Risk Elements.
6. Select Investments that together cost-effectively reduce Risks (thereby leading to attainment of the Requirements).

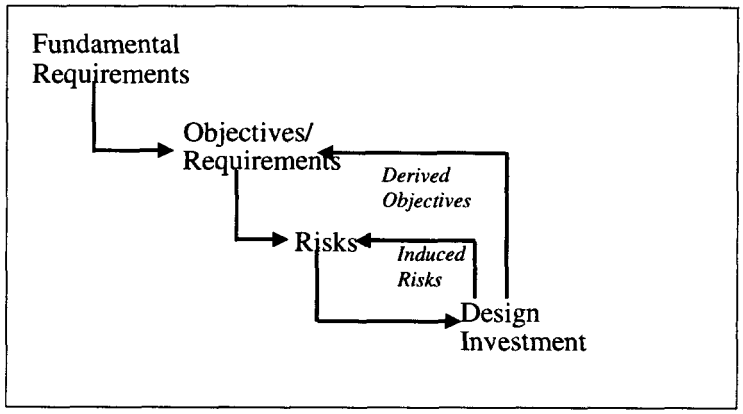

Figure 1:: Requirements Flow Down and Ripple Effects of Option

The DDP tool supports these steps. Its GUIs help users to enter, organize and edit the various kinds of information (Objectives, Effects, etc.). Quantitative calculations are performed automatically. For example, the magnitude of a Risk Element is computed as the product of its likelihood of occurrence (taking into account the reducing effects of investments) and its impact (sum of its impacts on the individual objectives). The overall purpose of DDP is to allow users to understand the often-complex interrelationships between Risks, Requirements and Investments, so as to guide their judicious selection of Investments. Further, it provides an optimization scheme that determines the optimal combination of Investments to employ for attaining a balance of risk and cost based on the preferences and constraints established by the decision maker.

Mission design using DDP is in fact an interactive process, sketched in Figure 1. Fundamental requirements are the starting point. The objectives of the project and lower level requirements are derived from these fundamental

requirements. The events that can lead to the non-fulfillment of the objectives or the risk elements are then identified. Design choices are made to reduce the identified risks. These design choices, in turn, may introduce new risks and/or derived requirements. Therefore the mission design process is more cyclic than hierarchical and it takes a few cycles to refine the initial design and produce an acceptable design. The mission design process is dynamic in nature, and the flexibility of DDP is critical to easily capturing these refinements and modifications as the design matures.

2.2 *Motivating Example (HEM)*

In order to demonstrate the concept explored in this paper, we present a simple, Hypothetical Example mission to Mars

Requirements Tree:

1: Get Sample
- 1.1: Go To MARS
  - 1.1.1: Launch
  - 1.1.2:Cruise
  - 1.1.3: EDL
    - 1.1.3.1: Land softly
    - 1.1.3.2: Land precisely
- 1.2: Dig the surface
- 1.3: Collect sample
- 1.4: Bring sample to earth
  - 1.4.1: Rendezvous with orbiter
  - 1.4.2: Return orbiter to earth

2: Collect interesting data
- 2.1: Go To MARS
  - 2.1.1: Launch
  - 2.1.2: Cruise
  - 2.1.3: EDL
    - 2.1.3.1: Land softly
    - 2.1.3.2: Land precisely
- 2.2: Find interesting spots
- 2.3: Take pictures

2.4: Send data to earth

Risks Tree:

1: Sample transfer chain failure
- 1.1: Navigational issues
- 1.2: Rendezvous failure
- 1.3: Sample transfer failure

2: Launch vehicle explodes
- 2.1: During launch
- 2.2: During Cruise
- 2.3: After rover has been seperated

3: Environment variability
- 3.1: Dust storm season
- 3.2: dense environment

4: Poor aero maneuvering

5: Insufficient Power
- 5.1: Battery runs out during EDL
- 5.2: Solar array failure during experimentation

Figure 2: Hypothetical Example Mission to Mars

(HEM). The objective of this mission is to send a rover to Mars to take interesting pictures and collect a sample and return to earth. The risk and requirements tree for this mission is as follow:

The requirements tree summarizes the objectives of the mission into getting a sample and collecting interesting data and returning the sample and pictures taken to earth. Note that these two requirements have some common sub-requirements. For instance, in order to meet both of these requirements, we first need to go to Mars. Going to Mars, in turn includes having a successful launch, cruise and Entry, Descent, Landing (EDL) phase. The risk tree highlights some of the failure modes for this mission. These modes include the explosion of the launch vehicle, insufficient power and a handful of other risks that can impact one or more of the items in the requirements tree. Now let's consider the effects of the risk items on the requirements. For instance, consider the "Environment variability" risk item. This has been classified into a "dust storm season effect" and a "dense environment effect". It can have an effect on items "Land precisely", "Get sample", and "Collect interesting data". Amongst the sub-items for the "Get Sample" item, the environmental variability can impact the items "Collect sample" and "Return Sample to Earth". The problem that we address in this paper is quantifying each of these effects with consideration of the dependencies that exist within and in between each of the risk and requirements categories. In the following sections, we identify each of these dependencies and show how they can be classified and dealt with accordingly.

3. *APPROACH*

In order to calculate the exact requirement attainment measure, we classify and quantify the dependencies within and in between the Requirements and Risks categories. In DDP, these categories are represented with a tree-like data structure. Figure 2 shows a screen shot of a sample risk tree in DDP.

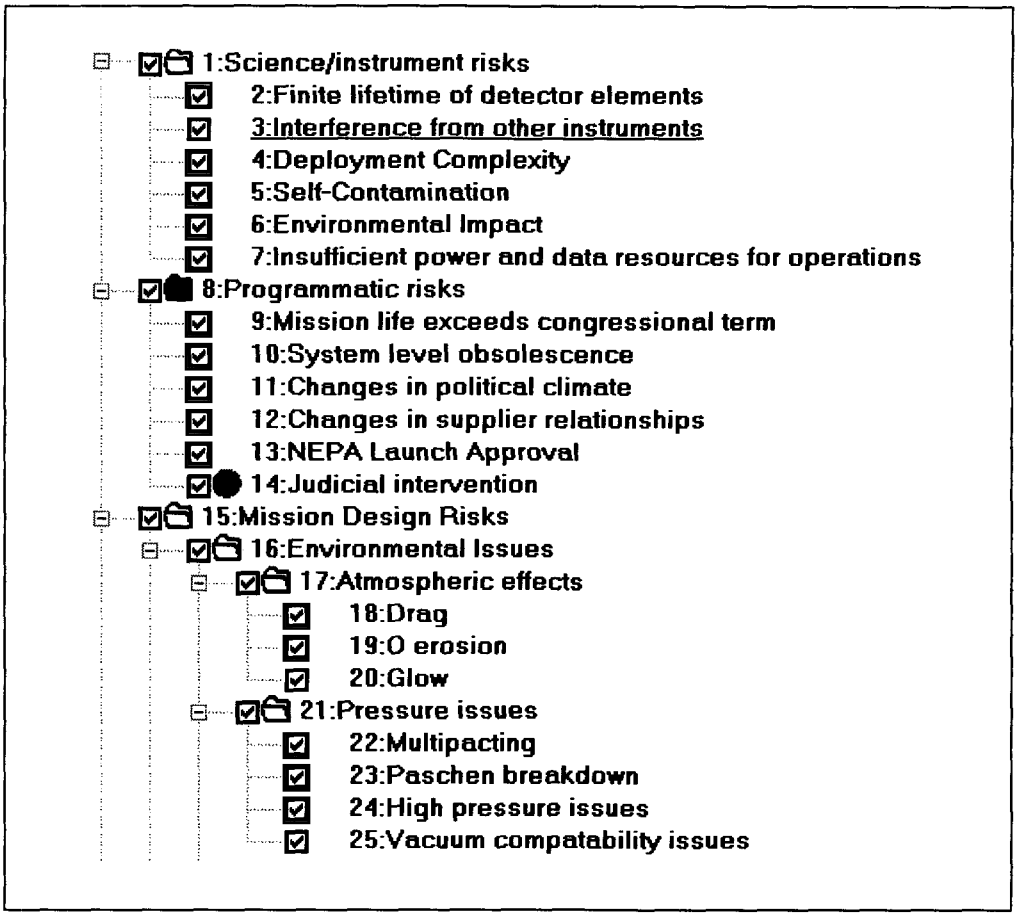

Figure 3: Example "Risk Tree" in DDP

Note that the data structures are not trees in a graph theoretical sense because they may contain loops as two nodes may share the same child. In DDP, including the same child under each shared parent shows this. Nevertheless, we borrow from the terminology used for trees in graph theory.

The goal is to obtain the exact requirement attainment based on the weight of each requirement, the effect of the risks on the requirements, and the probability of occurrence of the risk elements. Therefore, there are two main problems to be resolved. One is allocating the correct weight to each of the elements in the requirements category. In order to do so, we must carefully consider the dependencies that exist in this category. Let's consider the following definitions and measures:

- Intra-system dependencies:

Independent elements: Element isn't ancestor or descendent of other element, they don't share any children, and the occurrence of one does not effect the other.

Dependent elements: Elements can be dependent in two ways:

- *Relational Dependence*: Elements either have an ancestor-descendent relationship, or they share a parent or child.
- *Sequential Dependence*: The occurrence of one element depends on the occurrence of the other.

In our hypothetical example (HEM), the two main requirements "Get Sample" and "Collect Interesting Data" are relationally dependent elements since they share the "Go to Mars" tree. The elements "Launch" and "Cruise" are sequentially dependent since we cannot cruise unless we've already launched successfully.

- Inter-system dependencies:

Independent elements: Two elements from the same category that are not linked to any common elements (or any of its ancestor or descendents) in the other category.

Dependent elements: Two elements from the same category that are directly or indirectly linked to at least one common element in another category. An indirect link indicates that one or more of their higher or lower level elements (ancestor or descendents) are linked to the same element in the other category.

3.1 Aggregating requirement weights:

Now the question is how to allocate appropriate weights to each of the elements of the requirements tree based on the information that we elicit from the decision maker. It's realistic to assume that the decision maker allocates weights only to the highest-level parents of the requirements tree.

Given that level of information, the best we can do is to divide it equally between the children at each level. The problem is that these requirements may not be independent. If there are only relational dependencies between the requirements, it's reasonable to assume that the value of an element that appears under several requirements is the sum of the weights that have been allocated to it under each of the elements. So, in order to allocate appropriate weights to each element in the requirements tree we follow the following steps:

1. Elicit the value of each high-level parent node from the decision maker.
2. Divide it equally between its children.
3. For any child that has children:
    a. Divide its weight between its children.
    b. Continue until the depth of the tree is covered.

Using this simple approach, all siblings assume the same weight under the same parents. However, an element that appears more than once in the tree has a total weight equal to the sum of the weights it assumes under each of its parents.

If there are sequential dependencies between the different requirements, this indicates that we haven't broken down the requirement into it's different subsets, but we have specified the sequence of events that need to take place in order for the requirement to be satisfied. This information can be used to solve for the probability of satisfaction of the requirement using the probability of success (or failure) of the various risk elements that impact the sub-requirements. The solution techniques used for this purpose appear in the literature for phased mission system analysis [8] and are beyond the scope of this paper. However, the point to be addressed in this paper is that we don't allocate weights to sub-requirements that are sequentially dependent. In the case of HEM, there are values associated with each of the requirements "Collect interesting data" and "Get Sample", but none with their descendents.

In DDP, we provide the user with the capability to assign arbitrary weights to each descendent. However, if the user only provides weights to the higher-level requirements, the weights are equally distributed between the descendents as explained above.

The expected degree of requirement attainment is the weighted sum of the probability of attainment of each of leaves (or lowest level descendents) of a requirement.

Note that we are assuming that a requirement simply breaks down or partitions into its children. For more complex requirement tree relationships, this simple approach may not work.

3.2 Determining aggregate impacts:

The next issue to be resolved is finding the degree to which each requirement is satisfied based on the probability of occurrence of the risk elements and the effects of the risk elements on each of the requirements. There are several points to consider in the course of this endeavor.

In the context of DDP, we allow risk elements to have the following relationships with each other: A parent may be the logical AND or OR of two (or more of it's children.). In order to make sure that we don't double count the effect of a single risk element on a given requirement, we need to consider the various scenarios that might occur and find ways of dealing with them. First we consider the simple category of cases where the children are subsets of the parents.

Category One: Children are simple subsets of their parents.

The following cases may occur under this category:

1. A requirement may be effected by a descendent and an ancestor.

The issue is to ensure that we are not double counting the effect of the descendent by considering the effect of the ancestor at some level. To this effect we should start by matching the lowest level descendents to the requirements. If it turns out that some dependency is not covered at some level, that would indicate that the parent isn't properly partitioned into it's children and the union of the children does not equate to the parent. In that case, it would be necessary to define more children to cover the difference between the parent and it's children.

2. An ancestor and descendent on the requirements side are dependent from an inter-category perspective.

This indicates that an ancestor and descendent are mapped to the same risk element on the risk side. Note that they cannot be mapped to different generations of a risk element as we disallowed that in the previous clause. If the effect of that risk element on the ancestor is the same as its effect on the descendent, that means that we're probably double counting that effect. If the effect of that risk element on the ancestor is different, then again it indicates that at some level we haven't partitioned a parent into all its possible children and therefore need to add a child somewhere. In either case, we resolve this situation by disallowing such dependencies.

Therefore for this category, where the children are simple subsets of their parents at all levels, we disallow inter-category dependencies of two elements, from either the risk or requirements tree, that have an ancestor-descendent relationship.

Consider the hypothetical mission to Mars example demonstrated in section 2. In this example, the "environment variability" on the risk tree effects the "find interesting spots" on the requirements tree. It turns out that the children of "environmental variability", that include "dust storm season" and "dense environment" also effect the "find interesting spots" requirement, but their aggregate impact is less than the impact of their parent. This is due to the fact that the parent, "environmental variability" hasn't been partitioned into these two children and another dimension of environmental variability, which is "hazardous rocks", has been left out. In

order to resolve this situation, we add the "hazardous rocks" risk element to the children of "environmental variability" and match it's effect to the "find interesting spots" requirement. As a result, the aggregate effect of the children of "environmental variability" now equals to it's effect and therefore, we don't count the effect of the parent in the interest of avoiding the double counting.

Category Two: Parents on the risk side are AND or OR's of their children.

The following cases may occur under this category:

1. *A requirement may be affected by both the descendent of an OR gate and the OR gate itself.*

If a requirement is affected by a risk element, it will certainly also be affected by the OR of that risk element with any other element, based on the nature of OR gates. However, allowing both of these effects to be taken into consideration would result in double counting. Therefore, we disallow mapping an ancestor from the risk category to a requirement element, if any of its descendents have been mapped to that requirement. Therefore, if a descendent of an OR gate corresponds to an element in the requirements category, we disallow it's OR gate ancestor to correspond to the same element.

2. *A requirement may be affected by both the descendent of an AND gate and the AND gate itself.*

If a requirement is affected by an AND gate, this indicates that the occurrence of all it's children together will affect it. The effect of the individual children, in this case, isn't considered. Therefore, in this case, we disallow the effect of the children on the requirement, once the effect of the AND gate has been established.

3. *An ancestor and descendent on the requirements side are dependent from an inter-category perspective*

This case is similar to its corresponding case in category one and will be treated accordingly.

Consider the case where we have two different sources of power: solar arrays and batteries. The solar arrays are the primary source of power and the batteries are a backup source. The requirement "have sufficient power" will be affected by the risk element " power failure". The element, "power failure", is in turn the AND gate with the two children "solar array failure" and "battery failure". In this case, the parent, "power failure" affects the requirement, and the effect of its individual children is not apparent.

4. *CONCLUSIONS& FUTURE DIRECTIONS*

The results of this work are twofold. First it provides a clearer understanding of the relationship between the two categories of requirements and risks and clarifies the type and nature of the dependencies. Second, it establishes protocols for computing the exact requirement attainment measures based on the probability of occurrence of the basic risk elements.

In the future, we plan on looking the dependencies between the two categories mentioned in this paper, and the "mitigations" category. We further plan on investigating the correspondence between the requirement and risk tree in the context of DDP and event trees and fault trees in the context of PRA (Probabilistic Risk Assessment.)[2,3]

5. *ACKNOWLDGEMENT*

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration

REFERENCE

[1] M.S. Feather, S.L. Cornford, M. Gibbel. "Scalable Mechanisms for Goals Interaction Management", *Proceedings 4th IEEE International Conference on Requirements Engineering*, Schaumburg, Illinois, 19-23 Jun 2000, IEEE Computer Society, pp 119-129.

[2] NASA's Third Workshop for Probabilistic Risk Assessment Methods (PRAM-3) for Managers and Practitioners, given at the Embassy Suites Hotel, Arcadia, California, June 3-6, 2002.

[3] *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners*, version 1.1, prepared for the Office of Safety and Mission Assurance, NASA HQ, Washington, DC, August 2002.

[4] S.L. Cornford, J. Dunphy, and M.S. Feather: "Optimizing the Design of end-to-end Spacecraft Systems using risk as a currency", *IEEE Aerospace Conference*, Big Sky, Montana, 2002

[5] S.L. Cornford, M.S. Feather & K.A. Hicks. "DDP – A tool for life-cycle risk management", *Proceedings, IEEE Aerospace Conference*, Big Sky, Montana, Mar 2001, pp. 441-451.

[6] S.L. Cornford, M.S. Feather, J.C. Kelly, T.W. Larson, B. Sigal & J.D. Kiper: "Design and Development Assessment", *Proceedings, 10th IEEE International Workshop on Software Specification and Design*, San Diego, California, 5-7 Nov 2000, pp 105-204.

[7] S.L. Cornford: "Managing Risk as a Resource using the Defect Detection and Prevention process", *Proceedings, 4th International Conference on Probabilistic Safety Assessment and Management*, 13-18 September 1998, New York City, NY, International Association for Probabilistic Safety Assessment and Management.

[8] L. Meshkat, L. Xing, S.K. Donohue & Y. Ou. "An Overview of the Phase-Modular Fault Tree Approach to Phased Mission System Analysis" 1st Int. Space Mission Challenges for Information Technology, 2003. July 2003, pg. 393-398, JPL Publication03-13A, Jet Propulsion Laboratory, California Institute of Technology.

[9] Y. Akao. "*Quality Function Deployment*", Productivity Press, Cambridge, Massachusetts, 1990.

[10] M.A. Greenfield "Risk Management: 'Risk as a Resource' " http://www.hq.nasa.gov/office/codeq/risk/

BIOGRAPHIES

Steven L. Cornford, Ph.D.
MS 179-224
Jet Propulsion Laboratory,
California Institute of Technology, Pasadena, CA, 91109
Email: Steven.Cornford@jpl.nasa.gov

Steven Cornford is a Senior Engineer in the Strategic Systems Technology Program Office at NASA's Jet Propulsion Laboratory. He graduated from UC Berkeley with undergraduate degrees in Mathematics and Physics and received his doctorate in Physics from Texas A&M University in 1992. Since coming to JPL he focused his early efforts at JPL on establishing a quantitative basis for environmental test program selection and implementation. As Payload Reliability Assurance Program Element Manager, this evolved into establishing a quantitative basis for evaluating the effectiveness of overall reliability and test programs as well as performing residual risk assessments of new technologies. This has resulted in the Defect Detection and Prevention (DDP) process is the motivation for this paper. He received the NASA Exceptional Service Medal in 1997 for his efforts to date. He has been an instrument system engineer, a test-bed Cognizant Engineer and is currently involved with improving JPL's technology infusion processes as well as the Principal Investigator for the development and implementation of the DDP software tool.

Martin S. Feather, Ph.D.
MS 125-233
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
Email: Martin.S.Feather@jpl.nasa.gov

Martin Feather is a Principal in the Software Quality Assurance group at JPL. He works on developing research ideas and maturing them into practice, with particular interests in the areas of software validation (analysis, test automation, V&V techniques) and of early phase requirements engineering and risk management. He obtained his BA and MA degrees in mathematics and computer science from Cambridge University, England, and his PhD degree in artificial intelligence from the University of Edinburgh, Scotland. Prior to joining JPL, Dr. Feather worked on NSF and DARPA funded research while at the University of Southern California's Information Sciences Institute. For further details, see http://eis.jpl.nasa.gov/~mfeather

Leila Meshkat, Ph.D.
MS 301-180
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109
Email: Leila.Meshkat@jpl.nasa.gov

Leila Meshkat is a Mission Systems Engineer at the Mission & Systems Architecture Section of the Jet Propulsion Laboratory. Prior to joining JPL, she was a Research Associate at the University of Southern California's Information Sciences Institute and a Lecturer at the USC School of Engineering. She holds a PhD in Systems Engineering from the University of Virginia, an MS in Operations Research from the George Washington University and a B.S. in Applied Mathematics from the Sharif University of Technology. Her current research interests include Reliability and Risk Analysis and she is a member of IEEE and Omega Rohm.