Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface

Martin S. Feather
Jet Propulsion Laboratory, California Institute of Technology
Martin.S.Feather@jpl.nasa.gov

Abstract

Early risk assessment is key in planning the development of systems, including systems that involve software. Such risk assessment needs a combination of the following elements:

- *Severity estimates for the potential effects of failures, and likelihood estimates for their causes*
- *Fault trees that link causes to failures*
- *Efficacy estimates of design and process steps towards reducing risk*
- *Distinctions between preventing, alleviating and detecting (thereafter removing), risks*
- *Risk preventions that have potential side effects of themselves introducing risks*

The paper shows a unified approach that accommodates all these elements. The approach combines fault trees (from Probabilistic Risk Assessment methods) with explicit treatment of risk mitigations (a generalization of the notion of a "detection" seen in FMECA analyses). Fault trees capture the causal relationships by which failure mechanisms may combine to lead to failure modes. Risk mitigations encompass (and distinguish among) options to prevent risks, detect risks, and alleviate risks (i.e., decrease their impact should they occur).

This approach has been embodied in extensions to a JPL-developed risk assessment tool, and is illustrated here on software risk assessment information drawn from an actual project's software system FMECA (Failure Modes, Effects and Criticality Analysis). Since its elements are typical of risk assessment of software and its system interface, the findings should be relevant to a wide range of software systems.

Keywords: FMEA, FMECA, SFMECA, FTA, PRA, Risk-informed decision-making, Cost-benefit tradeoffs, Probabilistic Risk Reduction, Failure Modes

1. Introduction

Failure Mode and Effect Analysis (FMEA) is a commonly used approach to risk assessment for a wide variety of systems. Its origins trace back to Military Procedure MIL-P-1629 developed by the United States Military in 1949. When prioritization is emphasized [14], the method is often referred to as Failure Modes, Effects and Criticality Analysis (FMECA). For a relatively recent overview of their application to *software* systems, see [15].

The focus here is on the use of risk assessment *early* in the software development lifecycle. Early assessment is key in planning the development of systems, including systems that involve software. Major choices among design and development alternatives are made early in design, and should be guided by the risk-informed insights gained from risk assessment. FMECA seems well suited to such early lifecycle assessments.

This work is conducted in the context of spacecraft development, a setting exemplifying the challenges of safety/mission critical development of complex devices. Risk plays an obviously important role in such applications. Spacecraft software is particularly critical, because its failure can directly jeopardize the mission (e.g., software's role in Ariane V's demise [1], and as the most probable cause of loss of the Mars Polar Lander [12]). Furthermore, a key portion of the spacecraft software, the so-called "fault protection" system, is used to protect the spacecraft from all kinds of anomalies, both software and hardware in origin. In settings such as this there are design tradeoffs between the treatments of risks inherent in the spacecraft's mechanisms, the risks of defects in the spacecraft software, and the costs of options to reducing those risks (e.g., the cost of using more reliable hardware; the cost of a more thorough software V&V effort).

In early phases of spacecraft development it is common to apply some form of FMECA-like risk analysis to help identify and inform such design tradeoff decisions. A FMECA works well as a means to rapidly capture, represent and reason about much of the risk-related information at this stage in design. Using a FMECA it is possible to capture:

- Identification of the cause(s) and/or mechanism(s) of failure and likelihood estimates for them
- Severity estimates for the potential effects of failures
- Identification of measures planned for that reduce the risk, including their efficacy ("detectability")
- Identification of options to further reduce risk, and estimates of their efficacy (i.e., the reduced severity and/or likelihood that would result from their application)

However, some risk information goes beyond that which can easily be represented within the traditional FMECA framework. Some instances of such are as follows:

- Causal relationships between failure mechanisms and failure modes, indicating how occurrences of failure mechanism instances may combine to lead to failure modes. Fault-tree like structures are fundamental to representing and reasoning over such causal relationships.
- Distinctions between the risk reduction options. In particular, the ability to distinguish between preventing, alleviating, and detecting (thereafter removing) risks is needed to estimate the benefit (reduction of risk) and cost of a proposed design and its development plan.
- Risk preventions that have potential side effects of themselves *introducing* risks.

The purpose of this paper is to present and illustrate an approach that accommodates all these elements. This approach combines fault trees (from Probabilistic Risk Assessment methods [20]) with explicit treatment of risk mitigations (a generalization of the notion of a "detection" seen in FMECA analyses - www.fmeca.com). Fault trees capture the structural relationships by which failure mechanisms may combine to lead to failure modes. Risk mitigations encompass (and distinguish among) options to prevent risks (e.g., training; following of coding standards), detect risks (e.g., tests and analyses), and alleviate risks (i.e., decrease their impact should they occur, e.g., contingency mechanisms to recover gracefully from error states).

This approach has been embodied in extensions to a JPL-developed risk assessment tool. The approach is demonstrated on software risk assessment information drawn from an actual spaceflight project's software system FMECA. Since its elements are typical of risk assessment of software and its system interface, the findings should be relevant to a wide range of software systems.

The remainder of the paper is organized as follows:

Section 2 introduces the case study and the (traditional) FMECA representation of risk.

Section 3 describes a JPL-developed risk assessment method that represents information on risks, the objectives that those risks threaten, and the mitigations that are options for reducing those risks

Section 4 describes fault tree extensions to the aforementioned risk assessment method. Fault trees are used to represent the causal relationships between failure mechanisms and failure modes. The combination of fault trees with the explicit treatment of mitigations is the novel advance that makes possible the unified approach.

Section 5 shows how the FMECA information is represented using this extended risk assessment tool.

Section 6 illustrates the utility derived from having accomplished this representation.

Section 7 provides conclusions, status and directions for future work.

2. Case study

This section introduces a case study, the FMECA risk assessment for an actual spacecraft system. Note: for the purposes of this paper, identifying information has been deliberately suppressed so as to conceal sensitive information. A standard FMECA, populated by the spacecraft system personnel, is used to hold information on the spacecraft system's leading risks, and options for reducing those risks.is used. The FMECA structure is described first. An example row of the FMECA is then described in detail. Further elements of the FMECA are introduced in later sections of the paper.

2.1 Case study FMECA structure

The FMECA is a worksheet, downloaded from http://www.fmeainfocentre.com (the "FMEA Info Center"). This takes the form of a spreadsheet into which project-specific risk information can be entered. Its structure is as follows: rows are used to capture distinct failure mechanisms (causes); information on each such failure mechanism is organized into the following columns:

- **Item / Function** – a short label that serves to identify to the reader the system element involved.
- **Potential Failure Mode** – a brief textual description of the system-level failure that will potentially result.
- **Potential Effect(s) of failure** – a brief textual description of the consequences of that failure.
- **Severity** – on a scale of 1 (least) to 10 (most), how bad such a failure would be were it to occur.
- **Potential Cause(s) / Mechanism(s) of Failure** – a brief textual description of the cause of the failure.
- **Likelihood** – on a scale of 1 (least) to 10 (most), how likely such a failure is.
- **Current Design Controls** – already planned-for measures of the current design and its development that serve to reduce severities and/or likelihoods..
- **Detectability** – on a scale of 1 (*most* detectable) to 10 (*least* detectable), how well the current design controls are at "detecting" such failures prior to their actual occurrence. Note that a lower numerical score equates to a more effective detection; this is so that a simple multiplication is all that is required for the RPN calculation in the next column.
- **Risk Priority Number (RPN)** – the product of Severity, Likelihood and Detectability. The higher this calculated number, the greater the overall risk.
- **Recommended action(s), etc**. – further columns to use to list response plans, including the Severity, Likelihood and Detectability values that would result

from their application, and track their status (e.g., whose responsibility it is, when it is to be done by, whether action has yet been taken).

The entire FMECA contains 39 such data rows. Space limits preclude listing them all. The following subsection considers one of its rows in depth.

2.2 Case study FMECA – example row

An example data row from the spacecraft FMECA is shown in Table 1. The top row holds the column headers; the actual data is the second row. The first column of the original table (**Item/Function**) is deliberately omitted here, so as to conceal sensitive spacecraft design information. The trailing columns (**Recommended actions, etc.**) are also omitted, since they were empty in the original FMECA (at the time of its construction this information had not yet been ascertained). The columns shown and their data values are as follows:

- **Potential Failure Mode** – the failure causes identified in this row could trigger a *CPU Reset*
- **Potential Effect(s) of failure** – a CPU reset would cause loss of (in-core) information about the current state of the system, and require a complete reboot of the system.
- **Severity** – "*Low*" is actually in the middle of the scale, equating to a numerical value of 5. The textual guide for this rating scheme states "System inoperable without damage". The severity score of this failure is highly dependent on the intended use of the system. If this were a system that controlled a time-critical operation (e.g., controlled the entry, descent and landing portion of a spacecraft's mission), severity of a complete reboot would likely be rated very high, perhaps even catastrophic. For the case study system, it is not so drastic an event.
- **Potential Cause(s) / Mechanism(s) of Failure** – two very different causes are listed here, one related to electrical power to the computer, the other to a software problem.
- **Likelihood** – "*Low: Relatively few failures*" is towards the lower end of the scale, equating to a numerical value of 3, presumably because the power supply is expected to be relatively reliable, and software errors that would trigger a reset are thought to be unlikely.
- **Current Design Controls** – already planned-for measures in the current design call for the state information to be stored (meaning that it can be recovered), and for the system to await ground (i.e., communication from controllers back on Earth) following a reboot.
- **Detectability** – "*Almost Certain*" equates to 1, the lowest value on the numerical scale (i.e., the *most* effective possible kind of detection). In this example, storing state information on the fly is expected to almost certainly overcome the danger of loss of in-core state information, since it will be available for recovery from storage. Awaiting commanding from the ground is the planned way to re-initiate spacecraft operations. The original definition of "detectability" (from http://www.fmeaca.com) reads:

  > "Detection is an assessment of the likelihood that the Current Controls (design and process) will detect the Cause of the Failure Mode or the Failure Mode itself, thus preventing it from reaching the Customer."

  Observe that here the example row makes liberal use of "detectability" to encompass a recovery action rather than a preventative measure.
- **RPN** (Risk Priority Number) – the product of Severity, Likelihood and Detectability numbers is *15*, which is relatively low; the lowest possible such value is 1, while the highest is 1000.

In just this one row there are instances of the following phenomena: a severity ("*Low*") estimate for the potential effect of failure; a likelihood ("*Low; Relatively few failures*") estimate for the combined causes of that failure; expression of multiple causes that can lead to failure ("*Power surge or drop; Internal software error*"); estimated efficacy ("*Almost Certain*") of a design step adopted to reduce risk ("*Store state information & await commanding from ground*") by, it appears, reducing the severity of the outcome of the failure mode (as contrasted to decreasing its likelihood of occurrence, say).

The goal of this paper is to show how the FMECA information can be represented within a framework that allows for reasoning over the entire set of information.

Table 1. Sample Data Row from Case Study Software System FMECA

Potential Failure Mode	Potential Effect(s) of failure	Severity	Potential Cause(s) / Mechanism(s) of Failure	Likelihood	Current Design Controls	Detectability	RPN
CPU Reset	Loss of all state information & complete reboot	Low *(= 5)*	Power surge or drop; Internal software error	Low: Relatively few failures *(= 3)*	Store state information & await commanding from ground	Almost Certain *(= 1)*	15 *(5x3x1)*

3. DDP: a risk assessment method that makes objectives and mitigations explicit

The foundation for this work is a risk assessment method, with custom software support, that has been developed and applied at JPL and NASA. For historical reasons the approach is called "Defect Detection and Prevention" (DDP). This name reflects its origins as a method for quality assurance planning of hardware systems [2]. Since its origins DDP has evolved to support risk assessment for, especially, early design phases of spacecraft system and subsystems.

This section provides background information on DDP, focusing on its core features relevant to the focus of this paper. The section that follows will show the extension of DDP with logical fault trees, key to representing causal information.

The core DDP process deals with three key types of data: *Objectives*, *Risks* and *Mitigations*. Briefly:
Objectives (a.k.a. *Requirements* or *Goals*) are the things that the system is to achieve, and the limitations within which it must operate.
Risks (a.k.a. in the software realm, "*defects*" and "*bugs*") are all the kinds of things that, should they occur, would lead to failure to attain Objectives.
Mitigations are the actions that could be applied to reduce Risks.

Risks are connected to the Objectives they would detract from (were the Risk to occur), and to the Mitigations that reduce them (were that Mitigation applied). Figure 1 shows the "topology" of a DDP model of Objectives, Risks and Mitigations.

Detailed information is stored as attributes associated

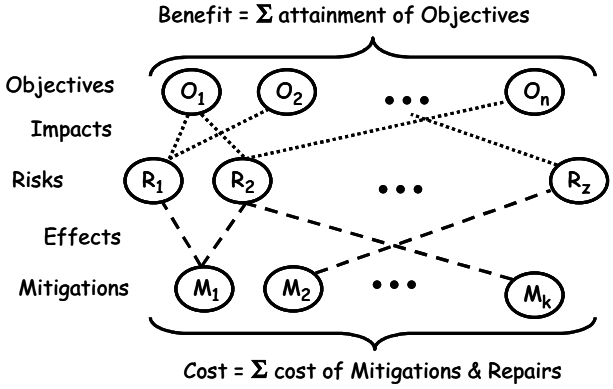

Figure 1. Topology of DDP model

with objects of these types. Attributes common to all three types include name, description, etc. Some key attributes for specific data types include:

An Objective has a "*weight*", used to represent its relative importance (some objectives are more important than others).

A Risk has an "*a-priori likelihood*", namely the likelihood of that Risk occurring if nothing is done to prevent it. A Risk also has a "*cost of repair*". This is the cost of repairing the problem (e.g., the cost of fixing a coding bug, or the cost of adding a missed requirement). In the software engineering community it is widely understood that these costs escalate through the course of the software lifecycle (e.g., the cost of correcting a flawed requirement at requirements time vs. at later phases in development).

A Mitigation has a "*cost*" (or costs), namely the resource costs of applying it. In our world of spacecraft development, there are typically several kinds of critical resources, e.g., budget ($), mass, volume, electrical power. A Mitigation also has a "*time*", typically the "phase" in the development effort at which it is applied (e.g., requirements time, design time, coding time). It is possible to use other time scales (e.g., financial quarters or, for long duration developments, years).

The DDP process deals with quantitative relationships that link Objectives, Risks and Mitigations, as follows:

Impacts are the quantitative relationships between Objectives and Risks, namely the proportion of the objective attainment that would be lost should the Risk occur. A risk can impact multiple Objectives to different extents, and similarly an Objective can be impacted by multiple risks, again to different extents.

Effects are the quantitative relationships between Mitigations and Risks, namely the proportion by which a Mitigation reduces a Risk should that Mitigation be applied. A Mitigation can effect multiple Risks, each to different extents, and similarly a Risk can be effected by multiple Mitigations, again each to different extents.

The key point is that a DDP model quantitatively connects Objectives, through the Risks that threaten them, to the Mitigation options for reducing those Risks (and thereby lead to increased attainment of Objectives). A DDP model specifies how to compute, for a selection of Mitigations, the level of Objectives' attainment, and the cost of those Mitigations plus the repair costs of the Risks whose correction is required. The DDP software performs these computations automatically.

The purpose of a DDP model is to support decision-making. In most practical situations, the total cost of all Mitigation options far exceeds the resources available, so DDP is often used to help guide judicious selection of mitigations. In some cases it becomes apparent that objectives are overly ambitious (given the limited resources available, and the risk averse posture of typical space missions), in which case DDP can be helpful at guiding descoping (strategic abandonment of Objectives). DDP has also proven useful for comparing two (or more) major design alternatives, by revealing the difference in their Risks (and the Mitigations that will be needed to sufficiently quell those Risks). In all these decision making applications, the DDP model helps by gathering on-the-fly

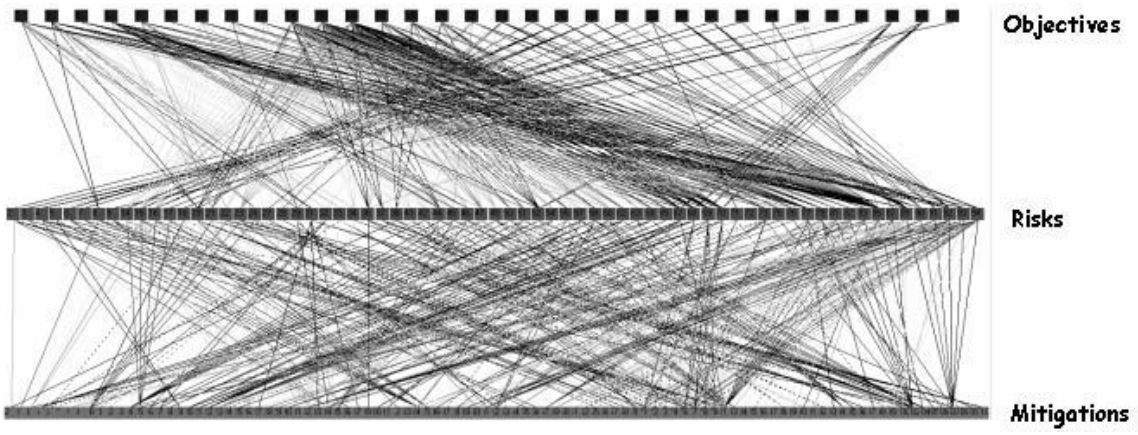

Figure 2. Topology of an actual DDP dataset data

knowledge from multiple experts and pooling that knowledge so as to help those experts derive insights from the knowledge as a whole. The DDP software supports knowledge gathering, representation, calculation and visualization. The amount of information taken into account in these DDP-supported risk studies is typically voluminous and highly coupled. This accounts for the need for an appropriate process and accompanying tool support. A sense of the detail can be seen from Figure 2, which shows the topology of the data in an actual DDP application.

Various aspects of DDP have been described in previously published papers: an early overview of its status and application may be found in [3]; a more recent description, including details of the underlying computational model, may be found in [7]. The extension of DDP with PRA concepts (so far some simple elements of fault trees) is more recent, so is not covered in those other publications. This extension is key to the ability to represent information such as found in the case study FMECA. The next section presents the extension.

4. Fault tree extensions to DDP

In a recent extension to DDP, *fault trees*, and the reasoning that goes with them, have been incorporated into the DDP representation of Risks. To date only the basic aspects of fault trees ("and" and "or" gates) are included. Incorporation of additional fault tree features is planned. In conjunction with DDP's other features, inclusion of the basic fault tree gates is sufficient to capture the information that occurs in the FMECA case study.

4.1 Background information

The goal of incorporating PRA's logical fault trees into DDP emerged following an earlier study in which probabilistic risk assessment (PRA) and DDP were separately applied to the same spacecraft design, and the results compared [5]. Briefly, the comparison showed DDP's relative strengths to be the ability to capture the wide range of risks that threaten a development, and to plan mitigations accordingly. It showed PRA's relative strengths to be the ability to faithfully represent the interplay of faults in combination, and to pinpoint areas of vulnerability in such combinations.

That study pointed to a *loosely* coupled way to integrate PRA and DDP, the essence of which is iteration between the two techniques. Start with DDP to rapidly pinpoint the riskier areas. Apply PRA to study them in with greater fidelity. Feed back the PRA results (likelihood and consequence into DDP), and re-rank the risks accordingly.

Such a loosely coupled integration is better than either technique alone, because it refines the accuracy of the risk assessment in the areas that matter the most. However further benefits emerge from a more intimate combination of the two techniques – it is the goal for such a unification of the two techniques that is the focus of this paper

4.2 Fault trees within the DDP topology

The key step is the inclusion of fault trees within the DDP topology of Objectives, Risks and Mitigations. As shown in Figure 3, the fault tree structures of PRA fit into the location where standard DDP has just single Risks. (Note: although standard DDP groups Risks into tree hierarchies, these serve only to *organize* them – akin to file folders in a directory structure).

4.3 Fault trees and DDP Objectives

Standard DDP "Impact" links connect Risks to the Objectives they threaten, using each link's quantitative measure to represent how much of the Objective would be lost were the Risk to occur. In the integration of PRA and DDP these same Impact links now connect nodes of logical fault trees (usually the root nodes – "top events" in PRA terminology) to Objectives. The probabilities of occurrence of these root nodes are calculated by means of the PRA techniques from the logical structure of the fault trees and the likelihoods of the leaf nodes of those fault trees. As will be explained shortly, DDP's "Effect" links come into play

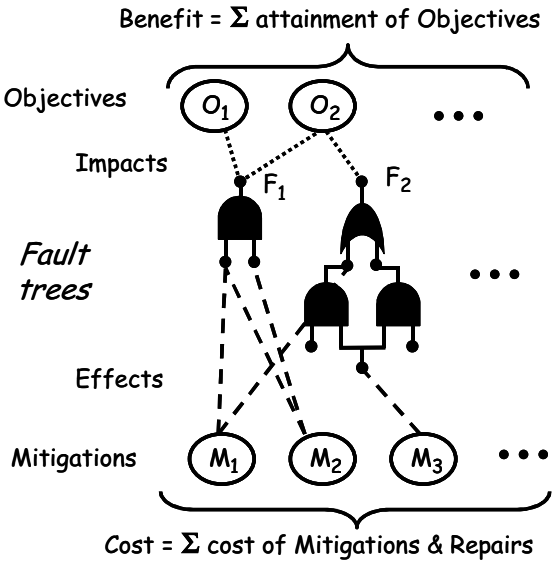

Figure 3. Topology of Fault Trees in DDP

to determine the likelihoods of those leaf nodes. Similar to standard DDP, an Objective may be "Impact" linked to multiple nodes of multiple fault trees, and a node of a fault tree may be "Impact" linked to multiple Objectives.

A hypothetical example of when it would be necessary to relate Objectives to *interior* root nodes of fault trees is as follows: consider a space mission with multiple science Objectives, one being a science experiment, another being the demonstration of a novel battery technology. The interesting case is when there is a standard battery to support the experiment, but the spacecraft design allows for the experiment to make use of the novel battery technology in the event that the standard battery fails. Thus the fault tree of risks to the science experiment would contain within it the subtree of risks to the novel battery. The root node of the no-power-available fault tree would be linked to the science experiment Objective, and the interior subtree of risks to the novel battery would be linked to the novel battery demonstration Objective.

4.4 Fault trees and DDP Mitigations

Standard DDP "Effect" links connect DDP Mitigations to the DDP Risks that they reduce (by decreasing the likelihood or decreasing the impact) or increase (for Mitigations that make some risks worse). In the integration of PRA's fault trees and DDP these same Effect links now connect DDP Mitigations to various locations within fault trees. The kind of Mitigation – prevention, alleviation or detection – determines the nature of the reduction, and constrains to which locations within a fault tree that Mitigation may be connected. Each of these mitigation types is discussed further in the subsections that follow. Similar to standard DDP, a Mitigation may be "Effect" linked to multiple fault tree locations (including within the same tree, and across different trees), and a fault tree location may be "Effect" linked to multiple Mitigations.

4.4.1 Prevention-type Mitigations and fault trees. "Prevention" type Mitigations can only be connected to *leaf* nodes ("basic events" in PRA terminology) of fault trees. Intuitively, this is because a non-leaf node of a fault tree correspond to logical combinations of that node's children. Hence the only way to affect its occurrence is to affect the occurrence of those children; applying this line of reasoning recursively, we see that this leads to affecting the occurrence of the leaf nodes (basic events) of the fault trees. Prevention mitigations serve to reduce the likelihoods, i.e., in PRA terminology, they decrease the "likelihood" half of the equation: risk = likelihood x severity.

See Figure 4 for a sketch of where they fit in to the picture of DDP with fault trees. Mitigation M is connected by an "Effect" link to a leaf node of the fault tree F. Recall that an "Effect" link has an associated quantitative value, indicating by how much the application of the Mitigation will reduce the risk. In this case, it indicates by how much the application of Mitigation M will reduce the *likelihood* of the basic event to which it is linked. By reducing the basic event's likelihood, the PRA calculation of the overall likelihood of F will be reduced, and so the expected amount by which F will detract from the Objectives to which it is linked will be correspondingly reduced.

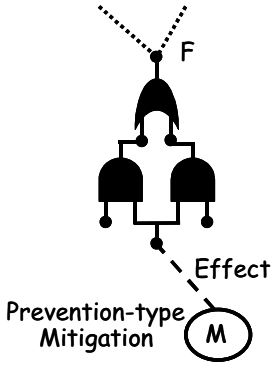

Figure 4. Prevention-type Mitigation and its effect on a fault tree

As a simple example of this, consider the development of a large software system for which security is one of the concerns. At the time of planning the development of the system, the project leads need to decide which, if any, training courses they should schedule their development team to take. A course on nework security could serve as a preventative measure – decreasing the likelihood that the programmers will make coding errors that contribute to security vulnerabilities (e.g., fail to check for buffer overflow). For an instance of a security study that employs fault trees to study vulnerabilities (e.g., [11]).

4.4.2 Alleviation-type Mitigations and fault trees. "Alleviation" type Mitigations are generally connected to the *root* nodes of fault trees, because it is the occurrence of these faults that detract from objectives attainment via the "Impact" links. Alleviation mitigations serve to reduce the impacts, i.e., in PRA terminology, they decrease the "severity" half of the equation: risk = likelihood x severity.

See Figure 5 for a sketch of where they fit in to the picture of DDP with fault trees. Mitigation M is connected by an "Effect" link to the root node of the fault tree F. Recall that an "Effect" link has an associated quantitative value, indicating by how much the application of the Mitigation will reduce the risk. In this case, it indicates by how much the application of Mitigation M will reduce the *severity* of the root node to which it is linked. By reducing the root node's severity, the amount by which F will detract from the Objectives will be correspondingly reduced.

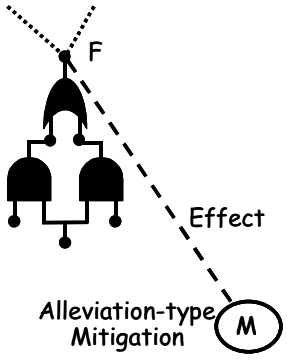

Figure 5. Alleviation-type Mitigation and its effect on a fault tree

The case of a non-root node of a fault tree linking to an Objective (recall "novel battery technology" example) would be an exception to this rule – it would make sense to link an alleviation-type Mitigation to that non-leaf node.

The DDP model assumes that an alleviation-type Mitigation's effect applies to the Risk (in standard DDP) or fault tree (in this extension of DDP) as a whole. Thus if a risk/fault tree is linked to several objectives by distinct "Impact" links, each of those impacts will be reduced by the same proportion. It is possible to imagine a more elaborate model in which alleviation-type Mitigations are linked to individual Impact links, allowing for the possibility that an alleviation-type Mitigation's Effectiveness differs from Impact to Impact.

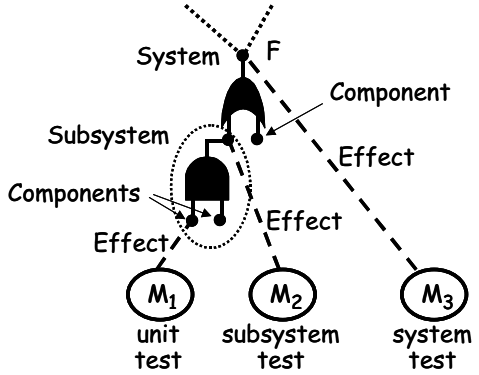

Figure 6. Examples of detection-type Mitigations and their effects on a fault tree

4.4.3 Detection-type Mitigations and fault trees. "Detection" type mitigations (e.g., tests, analyses, inspections) detect the presence of faults, which are then assumed to be repaired. In the DDP model extended with PRA's fault trees, a detection-type Mitigation is allowed to be connected to a node at *any* level of the fault tree. This is because a detection can be performed at any level.

For example, suppose the fault tree levels corresponded to a system, its subsystems, and components of those subsystems. A unit test, applied at the component level, is represented as a DDP detection-type Mitigation connected to the basic event node representing (kinds of faults) in that unit. A subsystem test is represented as a DDP detection-type Mitigation connected to the intermediate node representing that subsystem. A system test is represented as a DDP detection-type Mitigation connected to the root node representing the entire system. Figure 6 illustrates this scenario.

The way that detection-type Mitigations' effects decrease risk is as follows:

In the case of detection of a leaf node fault, the situation is straightforward – repair decreases the likelihood of that leaf node. When the standard PRA techniques are used to calculate fault tree likelihoods, they base their calculations on the decreased likelihoods that result from such repairs.

In the case of detection of a *non*-leaf-node fault (e.g., Figure 6's Mitigation M_2, a subsystem test) the situation is more interesting – repair equates to tracing to the cause(s) of that fault, namely the leaf nodes of the subtree, and repairing them (i.e., decreasing their likelihoods of occurrence). The net effect is that likelihoods of one or more leaf node faults are decreased, hence the likelihood of the overall tree containing those leaf nodes, when calculated by standard PRA techniques, is also decreased.

For example, suppose the mitigation is a system test, and the system is composed of two components, both of which must function correctly if the system is to function correctly – i.e., its fault tree would use an "or" node (a fault in of either component would produce a fault in the system). Faults discovered by a system test must result from faults in one or both of its units. If one of the units is more error prone than the other, then presumably the system test will reveal more errors attributable to that unit. DDP makes assumption of proportionality regarding this phenomenon – the proportion of errors of a more error prone unit will be a larger number of errors than the same proportion but of a less error prone unit.

The cost of repair is factored in when detection-type mitigations are involved, by multiplying the amount of each leaf node's likelihood reduction (i.e., the extent of the repair) by the unit repair cost for tht leaf node.

5. Application to case study FMECA

The previous sections have described the combination of logical fault trees from PRA combined with explicit treatments of risk reduction options, and how these are incorporated within the risk-assessment tool DDP. Use of this unified approach is now illustrated on the case study FMECA. The first subsection steps through an example, the representation of the FMECA's first data row (shown earlier). The second and third subsections describes in general terms the process of converting FMECA information into this form.

5.1 Case study FMECA's first row – representation in extended DDP

The sample row from the case study FMECA (shown earlier, in Table 1) is represented in DDP as shown in Figure 7. A column-by-column explanation of its representation follows:

- The potential failure mode "CPU Reset" becomes the root of a fault tree in DDP.
- Its potential effect(s) of failure "Loss of all state information & complete reboot" is prefixed with the word "Avoid" to become a DDP Objective.
- A DDP Impact link connects the two; the FMECA severity value of 5 in the scale [1..10] translates to a DDP value of 0.5 on that link.
- The potential cause(s) / mechanism(s) of failure "Power surge or drop; Internal software error" become two leaf nodes of the DDP fault tree, one related to power, the other to software, connected to the root node by an "Or" gate.
- The FMECA likelihood value of 3 in the scale [1..10] translates to a DDP a-priori likelihood value of 0.3 for the root of the fault tree. Since the fault tree consists of an "Or" gate with two leaf nodes, the same a-priori likelihood value is assumed for each of those nodes. To lead to a root node likelihood of 0.3, it can be inferred that the leaf node likelihoods are each approximately 0.16.
- The current design control "Store state information & await commanding from ground" becomes a DDP alleviation-type Mitigation.
- A DDP Effect link connect the Mitigation to the root of the fault tree; the FMECA detectability value of 1 in the scale [1..10] (where 1 = *greatest* detectability) translates to a DDP value of 0.9 on that link.

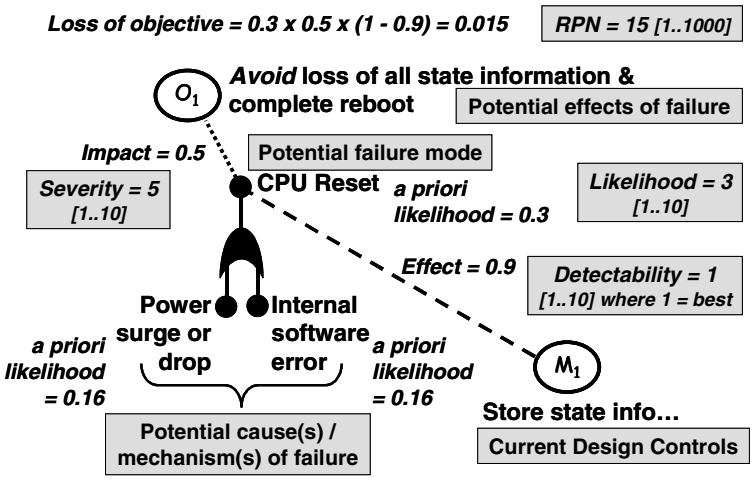

Figure 7. FMECA Sample Data Row in DDP

- The FMECA RPN value of 15 in the scale [1..1000] is computed by multiplying the FMECA Severity, Likelihood and Detectability. DDP itself computes a "Loss of objective" value by multiplying the fault tree root's likelihood (0.3), its Impact on the Objective (0.5), and (1 – the Mitigation's Effect value, i.e., (1 – 0.9)), yielding 0.015 as the result.

5.2 Representing FMECA information in extended DDP

The process of converting FMECA information into the equivalent representation within extended DDP is accomplished as follows:

- **Potential Failure Mode(s)** – each becomes the root of a DDP fault tree.
- **Potential Effect(s) of failure** – each becomes a DDP Objective. Since FMECA effects are *undesired* (whereas DDP objectives are *desired*), their names are prefixed with the word "Avoid" to form the name of the DDP Objective.
- **Severity** – a DDP Impact link whose value is the FMECA Severity value divided by 10 connects each pair of the above fault tree roots and Objectives.
- **Potential Cause(s) / Mechanism(s) of Failure** – these are organized into a fault tree below the potential failure mode. As seen in the example, some interpretation of the FMECA wording is needed to infer the structure of the fault tree.
- **Likelihood** – the root of the DDP fault tree is ascribed an a priori likelihood value computed by dividing the FMECA value by 10. Likelihood values for the leaf nodes of the fault tree are inferred from the fault tree root node's value. There may be many ways of doing this, so either some simple assumption should be made (e.g., all the leaf nodes' a priori likelihood values are identical, as was done in the example), or further

information sought from the experts who completed the FMECA.

- **Current Design Controls** – these become a single DDP Mitigation. If there are clearly several distinct design controls, it may be appropriate to split them into multiple DDP mitigations. The type of the DDP Mitigation (alleviation, detection or prevention) should be ascertained from an understanding of what the design controls are. In the case study, it was clearly an alleviation-type Mitigation.
- **Detectability** – a DDP Effect link whose value is (1 – the FMECA Detectability value) divided by 10 connects the above Mitigation(s) to the appropriate nodes in the fault tree.
- **Risk Priority Number (RPN)** – in the FMECA table, this is calculated as the product of Severity, Likelihood and Detectability. The higher this calculated number, the greater the overall risk. In DDP, the equivalent value is calculated by multiplying the fault tree root's a priori likelihood, its Impact on the Objective, and (1 – the Mitigation's Effect value). The higher the calculated value, the greater the loss of Objective due to the fault tree.
- **Recommended action(s), etc**. – further columns in the FMECA table (not shown above) used to list response plans provide information that can be represented as additional DDP Mitigations and Effect links, as needed.

Most of the FMECA's information is in a form amenable to conversion into the equivalent DDP representation. However, some aspects of the conversion require manual guidance, such as interpretation of the wording of the cause(s) / mechanisms of failure to infer the structure of the fault tree connecting them to the failure mode.

5.3 Representing dispersed but related FMECA information in extended DDP

In a typical FMECA, there may be information dispersed through the rows of the FMECA that is related. Extended DDP permits such situations to be represented, as follows:

5.3.1 Same FMECA element repeated in multiple rows. A cause/mechanism of failure listed in several rows of the FMECA would contribute several different failure modes, thus increasing the net benefit of inhibiting that cause/mechanism. In DDP this is represented by sharing the fault tree node representing that cause/mechanism among the several fault trees representing each of the failure modes.

A similar situation arises when a current design control occurs in several rows of the FMECA, with (possibly different in each case) "detectability" against several failure modes. In DDP this is represented as a single DDP Mitigation representing the design control, with multiple Effect links to different fault tree nodes.

5.3.2 Risk preventions that introduce risk. An interesting case of dispersed but related information is when a design control in one row of the FMECA is listed elsewhere as a potential failure mode or failure cause/mechanism. This is an instance of a risk prevention that has a potential side effect of *introducing* risk.

An example of this occurs in the case study FMECA: a "Watchdog Timer" is listed as a design control in one of the rows, and is listed in a different row of the same FMECA as a failure cause/mechanism. In its design control role, it works as an alarm clock to recognize when the system is in a deadlocked (or livelocked) state, triggering appropriate remedial actions. However, if it is incorrectly implemented or initialized (e.g., given too short a time period), it could trigger unnecessary and disruptive remedial actions.

Representation of this situation within DDP is accomplished by connecting the "watchdog timer" Mitigation via a DDP alleviation-type Effect link to the fault tree whose risk it reduces, and via a DDP likelihood *increasing* Effect link to the fault tree(s) for which inappropriate remedial actions triggered by the timer contribute to failure.

6. Utilizing the DDP representation

Having represented the FMECA information within DDP, it becomes possible to use DDP capabilities to scrutinize the overall risk situation, investigate mitigation alternatives, view the degree to which individual objectives are threatened by risks, etc. DDP's support for these activities includes automated calculation of risk levels and of values derived from them (e.g., for each Objective, the expected loss of attainment due to extant Risks), and interactive visualization mechanisms to present information to the user and permit on-the-fly "what if" studies. As illustration, this section presents several of DDP's visualizations, using the information from the case study FMECA as the information to display. See [7] for a more extensive description of DDP's capabilities.

6.1 Visualization of overall risk status

DDP's bar charts are commonly used to visualize the overall risk status. For example, the risk status of each of the 39 rows of the FMECA is shown in Figure 8, a DDP-generated bar chart with one bar per FMECA row. The black bars' heights represent the risk levels taking into account the risk-reducing effects of the identified design

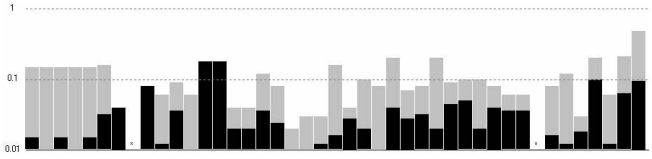

Figure 8. Risk status with/without mitigations

controls, while the grey bars' heights indicate where the risk would have been without those design controls. These can be sorted to draw attention the highest risks, as seen in Figure 9.

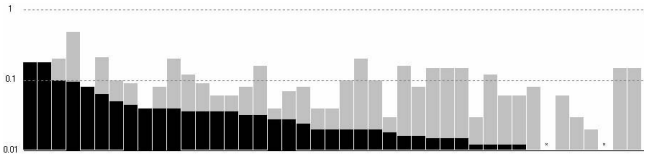

Figure 9. Risk status, sorted

An alternative DDP visualization of risk information is shown in Figure 10, where each risk is plotted on a 2-D chart, indicating its likelihood by position with respect to the vertical axis, and impact by position with respect to the horizontal axis. The chart is subdivided into different regions, the boundaries between which are lines of constant risk (which appear as straight lines because the chart axes are both log scale). The risk names are listed alongside for users' convenience – the figure here has been truncated so as to conceal sensitive information.

The various DDP displays are dynamically coupled – clicking on an element in one display causes the representation of that same element in *all* displays to be highlighted. For example, clicking on a bar in a bar chart of Risks causes that bar to be highlighted, causes the square corresponding to that risk in the 2-D chart to be highlighted, and causes the name of that risk in the listing alongside to be highlighted.

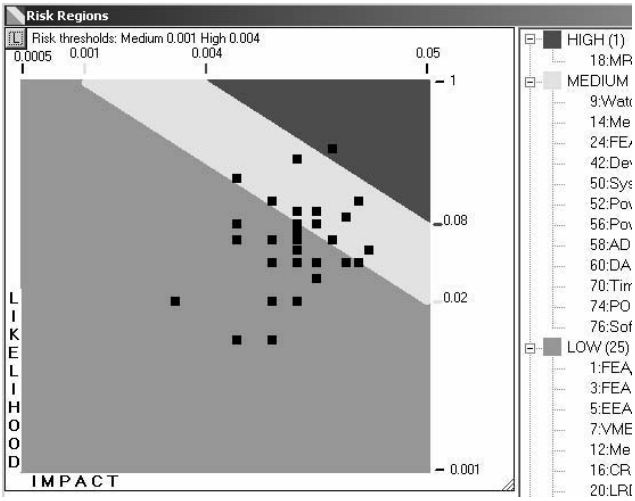

Figure 10. Likelihood/Impact risk status chart

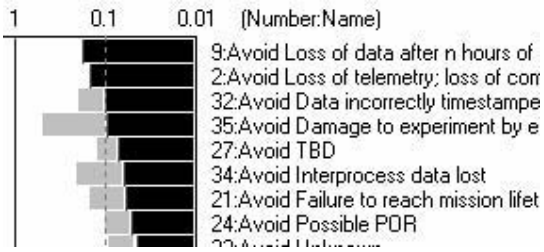

Figure 11. Objectives (sorted by risk status)

6.2 Visualization of overall objective status

DDP's inclusion of Objectives allows for scrutiny of their status – how important they are, and the extent to which their attainment is threatened by risks.

For example, the rows in the case study FMECA listed failure effects, representation of which was accomplished within DDP by prefixing their titles with the word "Avoid" to become DDP Objectives (e.g., "Avoid loss of data..."). Figure 11 shows a fragment of the DDP bar chart view of these Objectives, with their names listed alongside. As in figures 8 and 9, the black bars' lengths indicate risk taking into account the risk-reducing effects of the identified design controls, while the grey bars' lengths indicate where the risk would have been without those design controls. Risk for a given Objective is the sum total loss of attainment due to all the extant failure modes.

The Objectives' in Figure 11 are sorted sorted into decreasing order of how much they are threatened by risks. For variety, this figure shows DDP's "horizontal" bar chart layout, alongside which the index numbers and names of are listed.

6.3 "What if" studies and optimizations

DDP enables rapid investigation among alternative selections of potential risk mitigations (in the FMECA case study among design controls and recommended actions). The user may turn individual Mitigations "on" and "off" through the DDP interface. Each time this is done, the risk status is recomputed automatically, and the displays updated. For a dataset of the size of the case study FMECA, this takes under a second, so can be used for very rapid "what if" studies.

The visualization mechanisms can be set to reveal the risk changes that result from such explorations. For example, Figure 12 shows a bar chart display highlighting the risk *changes* in going from a selection of mitigations in which the watchdog timer design control is selected, to one in which it is unselected. The darker grey bars indicate risks that have increased, while the lighter grey bars indicate

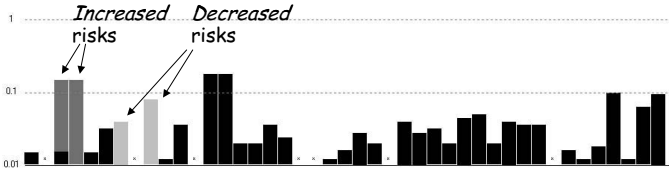

Figure 12. Visualization of changes to risks

risks that have decreased (greyscales are used for figures in this paper; in the tool, color can be used).

DDP calculates several summary measures of the Risk-reducing effects of each Mitigation. One such measure is of each currently unselected Mitigation's risk-reducing contribution with respect to the current selection of Mitigations. The measure calculates by how much total risk would be decreased were the Mitigation in question to be selected. Calculating this for each of the unselected Mitigations reveals the delta improvement that each would confer. DDP uses a simple bar chart to cogently display these results. These summary measures are analgous to (indeed, motivated by) the risk importance measures found in PRA. For example the Risk Reduction Worth *"… is a measure of the change in risk when a basic event …is set to zero. It measures amount by which risk would decrease if the event would never occur…"* [17]. While PRA measures relate to risks, DDP can also offer measures that relate to Mitigations, since these are represented explicitly within the DDP model.

When cost information is available, it becomes possible to compute the *cost* of a selection of mitigations as well as benefit (i.e., reduction of risk, leading to increased attainment of Objectives). This allows exploration of cost/benefit tradeoffs in the selection of risk mitigations. DDP has built in a heuristic search capability to locate near-optimal mitigation selections (e.g., for a limited amount of funding, find the selection of mitigations that will minimize risk). This is discussed further in [4]. Capabilities for allowing users to better understand their options in such circumstances are discussed in [8]. The case study FMECA does not (yet) contain cost information for the design controls, nor "recommended actions" to consider in addition to the identified design controls, so the data is not available to serve as illustration here.

7. Conclusions

This paper has presented a novel combination of logical fault trees with explicit treatment of risk mitigation options and explicit treatment of risk impacts against objectives. The goal is risk representation and reasoning encompassing all of the features seen in practical applications of risk-based decision-making early in the design of safety- and mission-critical systems.

This combination is fully implemented. Shown herein is a real-world example of its application to a spacecraft system's FMECA. Two ways of using this are possible: designers could continue to use simple risk tools such as FMECAs and, when appropriate, transfer their information to more elaborate risk tools for elaboration and deeper scrutiny; or, they could make the switch to those more elaborate tools..

Further motivation for an approach like this can be found in [13], where a combination of forward search (SFMEA) and backward search (SFTA) is advocated. Motivation for techniques able to take into consideration distinctions between defect prevention, detection and correction when assessing software are to be found in [9], and use Bayesian Belief Networks to combine knowledge of development process structure, etc. is in [10].

Closely related is the goal-tree refinement work in [19], wherein "goals" are the equivalent of DDP Objectives, and "obstacles" are the equivalent of DDP Risks; a quantitative treatment of goals is being added (until recently, goal satisfaction was purely binary). It seems that DDP's explicit treatment of Mitigations could be blended into this framework in a manner similar to the approach to blending with PRA's logical fault trees shown in this paper.

7.1 Status

The unified approach described has been implemented within the extended DDP software, available from http://ddptool.jpl.nasa.gov. All the information within the case study FMECA has been entered into this extended DDP. The charts shown in section 6 are screenshots taken from DDP in operation on this information. As a consequence, the risk implications of the entire FMECA information can be studied. The effects of altering selections of FMECA design controls (risk mitigation options) can be computed, and the results presented to users via cogent visualizations.

The overall process of converting the FMECA information into the equivalent DDP representation is partially automated. A macro performs the bulk of the conversion (e.g., the potential failure mode listed in each row of the FMECA is translated into a root node of a DDP fault tree). Once within DDP, the user must rearrange the information accordingly to capture the nuances of the system in question (e.g., to construct the logical fault trees that represent the causality implied by the FMECA's textual descriptions).This is not a large burden, hence it is feasible to continue to use the FMECA as the primary data-gathering tool, and update the DDP information when needed.

7.2 Future work

The next step in *application* of this work is to further its use in spaceflight design and development efforts. The mission whose spaceflight system FMECA served as this paper's case study was cancelled, so there is need to find an alternative mission on which to conduct such studies.

Possible next steps in *extension* of this work would be to consider incorporation of additional features into the unified risk representation and reasoning. One obvious source of such is to continue to draw from Probabilistic Risk Assessment – for example, PRA's use of dynamic fault trees to represent sequence dependencies in the temporal evolution of a system's behavior, e.g., [6]. Another avenue

to pursue is to elaborate the computation of the "value" of a given design. For example, [16] uses a utility model to compute sum total expected science return of alternative rover designs for operation on Mars.

The planned next step in *implementation* of this work is to collaborate more closely with an existing PRA tool to develop automated information exchange between DDP and the PRA tool, so as to utilize the latter's advanced capabilities (e.g., Markov models for solving elaborate fault trees). Collaboration towards this end is ongoing with J. Dugan and K. Sullivan of Univ. of Virginia with their PRA tool "Galileo" for dynamic fault trees [18].

8. Acknowledgements

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

Special thanks are due to DDP lead Steven Cornford (JPL), Stephen Watson (JPL) for providing the FMECA, and for fruitful discussions with Joanne Dugan (Univ. of Virginia), James Kiper (Miami Univ., Oxford OH), Leila Meshkat (JPL), Todd Paulos (JPL) and Andrew Shapiro (JPL).

9. References

[1] Ariane 5 Inquiry Board. "ARIANE 5 Flight 501 Failure", 1996.

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

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

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

[5] Cornford, S.L., Paulos, T., Meshkat, L. and Feather, M.S. "Towards More Accurate Life Cycle Risk Management Through Integration of DDP and PRA". Proceedings *IEEE Aerospace Conference*, Big Sky MT, Mar 2003.

[6] Dugan, J.B. and Assaf, T.S., "Dynamic Fault Tree Analysis of a Reconfigurable Software System", *`9th Int. System Safety Conference*, Hunstville, Alabama, Sept. 2001

[7] Feather, M.S. and Cornford, S.L. "Quantitative risk-based requirements reasoning", *Requirements Engineering* (Springer), Vol 8, No 4, 2003 pp. 248-265 – published online 25 February 2003, DOI 10.1007/s00766-002-0160-y.

[8] Feather, M.S., Kiper, J.D. and Kalafat, S. "Combining Heuristic Search, Visualization and Data Mining for Exploration of System Design Spaces", to appear in *INCOSE* 2004; advance copy available from http://eis.jpl.nasa.gov/~mfeather/Publications.html

[9] Fenton, N. and Neil, M. "A Critique of Software Defect Prediction Research", *IEEE Transactions on Software Engineering* 25(5), 1999.

[10] Fenton, N., Krause, P. and Neil, M. "A Probabilistic Model for Software Defect Prediction", To appear in *IEEE TSE* – contact Fenton at: norman@dcs.qmw.ac.uk

[11] Helmer, G., Wong, J., Slagell, M., Honavar, V., Miller, L. and Lutz, R.. "A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System", *Requirements Engineering* 7(4), pp. 207-220, 2002.

[12] JPL Special Review Board. "*Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions*", March 2000, JPL D-18709, Jet Propulsion Laboratory, California Institute of Technology.

[13] Lutz, R. and Woodhouse, R. "Requirements Analysis using Forward and Backward Search", *Annals of Software Engineering, Special Volume on Requirements Engineering*, 3, pp. 459-475, 1997.

[14] Moriguty, S. "*Software excellence: A total quality management guide*." Portland, OR. Productivity Press, 1997.

[15] Pennti, H. and Atte, H. "*Failure Mode and Effects Analysis of Software-Based Automation Systems*" STUK-YTO-TR 190 available from http://www.fmeainfocentre.com/download/softwarefmea.pdf

[16] Smith, J.H., Wertz, J., and Weisbin, C. "Building a Pathway to Mars: Technology Investment for Science Return," *Journal of Space Mission Architecture (JSMA)*, 3, Sep. 2003.

[17] Stamatelatos, M., et al. "Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners" http://www.hq.nasa.gov/office/codeq/doctree/praguide.pdf Office of Safety and Mission Assurance, NASA Headquarters, Washington, DC 20546, Aug 2002.

[18] Sullivan, K.J., Dugan, J.B. and Coppit, D. "The Galileo Fault Tree Analysis Tool", *Proceedings of the 29th International Conference on Fault-Tolerant Computing* (FTCS-29), 1999.

[19] A. van Lamsweerde & E. Letier. "Integrating Obstacles in Goal-Driven Requirements Engineering", *ICSE98 – 20th International Conference on Software Engineering*, IEEE-ACM, Kyoto, April 1998.

[20] Vesely, W.E., Goldberg, F.F., Roberts, N.H. & Haasl, D.F., "*Fault Tree Handbook*", U.S. Nuclear Regulatory Commission NUREG-0492, 1981.