OVECOMING OBSTACLES TO THE EXCHANGE OF INFORMATION BETWEEN RISK TOOLS

Martin S. Feather, Steven L. Cornford, Leila Meshkat, Luke Voss
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099
{Martin.S.Feather, Steven.Cornford, Leila.Meshkat, Luke.Voss}@jpl.nasa.gov

ABSTRACT

Risk tools have matured to the point where they support a variety of risk-informed decision making. While we may expect continued development of such tools to lead to further individual improvements, we believe the greatest gains are to be had from their integration.

Our work to date in connecting risk tools has had successes, but also has revealed there to be significant impediments to information exchange between them. These impediments stem from the well-known phenomenon of “semantic dissonance” [Kent, 1978] – mismatch between conceptual assumptions made by the separately developed tools. This issue represents a fundamental challenge that arises regardless of the mechanism of information exchange.

This paper explains the issue and illustrates it with reference to our experiences to date connecting several risk tools. We motivate this work, present and discuss the solutions we have adopted to surmount these impediments, and the implications this work has for future efforts to integrate risk tools.

1 Introduction

There exists a plethora of tools and techniques that support a variety of risk informed decision making. Some of this variety stems from independent realizations of *similar* approaches to solving similar problems (e.g., the similar approach to risk assessment seen in tools for Probabilistic Risk Assessment (PRA) such as Saphire, Cafta, QRAS, Galileo), some from *alternative* approaches to similar problem areas (e.g., the *quantitative* approach as exemplified by the aforementioned PRA tools, vs. the *qualitative* treatments of risk seen in some tools that support FMEAs and FMECAs), and some from addressing different problem areas (e.g., detailed risk *analysis* for a given system’s design as exemplified by the PRA tools vs. extensive support for ongoing risk *management* of large projects, e.g. as provided by Active Risk Manager).

While we may expect continued development of such tools to lead to further individual improvements, we believe at this point the greatest gains are to be had from their integration:

- For tools offering similar approaches to similar problem areas, integration that facilitated the exchange of risk information between those tools would enable the strengths of a given tool (e.g., one tool might offer a more efficient algorithm, another a more cogent visualization of results) to be brought to bear when appropriate, and would allow users to each stick with the tool they are familiar with while cooperating with one another.
- For tools offering alternative approaches to similar problem areas, integration that facilitated the sharing of the information they have in common would allow the parallel investigation of the problem with each approach, and comparison of their results; another advantage would stem from seamless transfer of information between risk tools that are suited to different phases of the lifecycle (e.g., having used a very agile but low-fidelity risk tool to make some initial down-selects among design options, it would be convenient to pass on the its risk analysis to serve as the starting point for more refined risk analysis in the next phase in development).
- For tools that address different problem areas, integration would allow their combination of strengths (e.g., a risk *analysis* tool would help in the risk informed decision making steps, while a risk *management support* tool would help track schedule and progress of the decisions made, and predictions forecasted).

Integration is clearly desirable. This issue is cogently presented in [Throop et al, 2005] for the closely related field of "…*automated reasoning tools [that] must represent graphs of causally linked events. These include fault-tree analysis, probabilistic risk assessment (PRA), planning, procedures, medical reasoning about disease progression, and functional architectures* …" The authors describe the goal of a comprehensive interchange format for causal phenomena. They point to the availability of mechanisms for representing and manipulating such interchange formats (e.g., they refer to XML-based representations leading towards RDF or OWL, and to the Reference Information Model, RIM [RIM], a meta-standard for specifying data formats). They also point to several groups in the process of developing standard representations for fault trees and/or risks (e.g., Standard for the Exchange of Product model data, or STEP (ISO-10303) [STEP]).

We share this desire for integration, and over the past two years have been involved in some preliminary steps to integrate several risk tools and techniques. We embarked upon a long-range vision of a unified framework within which risk tools could co-exist, exchange information, and be operated in a coordinated fashion. This was to support the design of spacecraft and space missions, extending from their early (conceptual) design phases onwards. The framework (called the "Risk Tool Suite for Advanced Design" – RTSAD) was built around an internal representation of risk information, to be capable of representing (and therefore facilitating the exchange of) multiple risk tools' various kinds of risk data. This effort sought to utilize existing risk tools, rather than recreate them from scratch, thus leveraging the significant development effort that had already been invested in each of those tools. However, because those tools had been developed separately from one another, several significant impediments emerged. One class of impediments in particular, the focus of this paper, stem from the well-known phenomenon of "semantic dissonance" [Kent, 1978] – *mismatch between conceptual assumptions made by separately developed tools*. Mismatches can range from the mundane (easily dealt with, provided that you recognize them! – e.g., different units of measurement, different data representations) to the complex (e.g., PRA tools manipulate structures such as logical fault trees, whereas other tools deal with risks as "atomic" objects). Identifying and surmounting these semantic dissonance impediments is a key step towards the goal of unifying multiple risk tools.

We found that a promising way to make progress towards this vision was to selectively build connections between risk tools so that they can exchange information. This had the advantage of enabling their combined strengths to be applied immediately to risk analysis problems at hand, and guided progress towards the creation of a standard representation for risk exchange in the broader integration setting. We were following this approach in parallel with the construction of the infrastructure of the RTSAD framework effort mentioned above. While that effort is no longer ongoing, our work to connect risk tools continues. Another advantage of our more selective approach to risk tool integration is that it narrowed the scope of the impediments stemming from semantic dissonance. The remainder of this paper discusses several of the impediments that we encountered, the solutions we adopted to surmount these impediments, and the implications this work has for future efforts to integrate risk tools. The paper is structured as follows:

Section 2 provides an overview of four risk tools that we have built connections among, and use as illustration throughout the paper.

Section 3 describes our motivations for seeking to connect these risk tools, and goes on to describe the kinds of impediments we encountered, stemming from information clashes attributable to semantic dissonance.

Section 4 presents the range of solutions we found the need to apply in order to solve the above impediments.

Section 5 illustrates the utility we achieved from our risk tool integrations.

Section 6 concludes the paper with a summary discussion, and our plans for future work.

2 Overview of the Risk Tools used in this study

The specific risk tools whose connection we discuss herein are as follows:

1. A spreadsheet-based Failure Mode and Effect Analysis (FMEA) tool from http://www.fmeainfocentre.com (the “FMEA Info Center”).
2. “Galileo”, a dynamic fault tree analysis tool that calculates system reliabilities.
3. “Risk and Rationale Assessment Program” (RAP), an agile, multi-user risk collection tool developed and used at JPL for collecting expert opinions about risk from designers involved in concurrent design sessions.
4. “Defect Detection and Prevention” (DDP), a risk analysis tool developed and used at JPL for risk assessment and risk mitigation planning.

Since we developed RAP and DDP ourselves, we were able to extend/adjust these two as the need arose. In addition, the leaders of Galileo (Professors Joanne Dugan and Kevin Sullivan, of the University of Virginia) were party to our integration efforts, and in response to our requests were willing to make interface adjustments to their implementation. The spreadsheet FMEA tool we simply used as-is.

In the subsections that follow we describe the salient aspects of each tool from the perspective of risk tool integration. It is important to note that our descriptions are not intended to convey the breadth and depth of the capabilities that each of these tools proffer, and indeed fall far short in this respect. We recommend readers pursue the references for more thorough descriptions of the tools.

2.1 FMEA tool

The FMEA tool 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).

This worksheet assigns numerical values (1-10) for Severity, Likelihood and Detectabilty values, but these are ordinals rather than cardinals. Thus these are *qualitative* measurement scales. For example, the upper end of the Severity scale is defined as

> 10: Hazardous without warning – “Very high severity ranking when a potential failure mode affects safe system operation without warning”

9: Hazardous with warning – "Very high severity ranking when a potential failure mode affects safe system operation with warning"

8: Very High – "System inoperable with destructive failure without compromising safety", etc.

The tool is to be found on the "FMEA Info Center", http://www.fmeainfocentre.com, along with many more resources on use of FMEAs, etc.

2.2 Galileo

The field of Probabilistic Risk Assessment (PRA) has developed tools and techniques to assess risks within complex systems. The key idea of PRA is to deduce the reliability of a system from knowledge of the system structure and knowledge of the reliability of the individual components from which the system is composed. Application of PRA techniques yields an overall assessment of a system's reliability, confidence measures of that assessment, and insight into the key vulnerabilities of that system, thus indicating areas most in need of improvement. PRA is now applied to a wide variety of systems; NASA has developed training course and associated material [NASA PRA, 2002] to assist its application to space missions.

There are many PRA tools; for our studies to date, we have worked with one such tool, Galileo, a software tool for Dynamic Fault Tree (DFT) analysis [Dugan et al, 1992]. Dynamic fault trees extend traditional fault trees in a way that makes them well suited for the analysis of, for example, computer-based systems. The DFT methodology uses special purpose gates to model sequential behaviors, such as functional dependencies, shared pools of spares, cold and warm spares and other time- and order-dependent events. Our integration studies to date have concentrated on only the simpler aspects of fault trees – the core features of "and" and "or" gates (see [Vesely et al, 1981] for the canonical description of fault trees). The essence is as follows:

- A **fault tree** comprises one or more nodes organized into a tree structure (strictly speaking, a directed acyclic graph with a single root, since it is permissible for an intermediate or leaf node to occur at multiple places within the tree.
- Each non-leaf node of a fault tree has a type, either an "**and**" node or an "**or**" node, the former (latter) indicating that the event represented by the node will occur if and only if all (at least one) of its child node events occur(s). (In most PRA tools, Galileo included, there are additional types of nodes, which our integration studies to date have not dealt with).
- Each leaf node of a fault tree has a **likelihood** assigned to it; the likelihoods of the non-leaf nodes are computed from knowledge of the likelihoods of their descendants.
- The root node of a fault tree has a consequence – the adverse impact on the system should the event represented by the root of that fault tree occur.

For more information on Galileo, see [Sullivan et al, 1999] to learn more about the application of Galileo and the kind of dynamic fault trees it supports. The implementation of Galileo itself represents an interesting approach to the software engineering of such tools, including low-cost pathways to their construction [Dugan et al, 1999] and formal validation [Coppit & Sullivan, 2003].

2.3 Risk and Rationale Assessment Program (RAP)

JPL uses a concurrent engineering team to analyze the feasibility of mission ideas and produce conceptual mission designs. The team consists of up to 20 engineers, each representing a different discipline, and a team leader. The RAP software tool was developed for this team environment, to allow the various designers to consider and communicate about the risks of their mission designs. It is a distributed system that enables the gathering, communication and consolidation of risk information. The individual risk elements that RAP users create and exchange each has the following attributes:

- **Title –** a concise name with which to refer to the risk
- **Description –** free-form text

- **Owner** – the team member who owns the risk (RAP allows for the different team members to each have their own assessment of a risk element; in RAP parlance, “risk element” is the shared risk concept, which is coupled to multiple “risk factors”, one per team member)
- **Objective –** the objective threatened by this risk were it to occur (defaults to “mission success”)
- **Likelihood**, expressed as a number on a scale of 1 (least) to 5 (most), where an element of this scale means a probability range (e.g., “5” means the probability range 0.7 – 1.0, “4” means the probability range 0.5 – 0.7, etc)
- **Impact**, expressed as a number on a scale of 1 (least) to 5 (most), where an element of this scale means a range of the proportional loss of objective (e.g., “5” means proportional loss in the range 0.7 – 1.0, “4” means the proportional loss in the range 0.5 – 0.7, etc.)
- **Mitigation –** an option that, if chosen, reduces the likelihood and/or impact of the risk (e.g., use of redundant design elements decreases likelihood of a system failure risk); the risk reduction is indicated by providing the risk’s likelihood and impact values that would be realized were that mitigation chosen.
- **Event(s) –** records the mission events (also defined through the RAP tool) correlated with this risk. This information will be useful for conducting more detailed Probabilistic Risk Assessment studies of the mission design.

Further attributes are used to track the risk’s author, its status (e.g., a risk may have been suggested, but not yet assessed), etc.

For further information on the RAP tool and its applications, see [Meshkat et al., 2003] and [Meshkat & Oberto, 2004].

2.4 Defect Detection and Prevention (DDP)

DDP is another JPL-developed risk tool, whose primary aim is to assist in planning cost-effective ways to reduce risks. DDP is suited to situations where are many options for reducing risks, yet resources (budget, time, and physical resources of the system itself such as mass, volume, electrical power) are limited, meaning only a fraction of those options can be afforded. The elements that DDP manipulates are:

- **Objectives –** the goals that the system being studied is intended to achieve.
- **Risks –** the events that, should they occur, will detract from attainment of (some) objectives
- **Mitigations –** the options available for reducing risks.

Each of the above has a **Title** and **Description**. In addition:

- Each objective has a “**weight**” – a non-negative number indicating its importance relative to other objectives (e.g., an objective with weight 10 is 5 times as important as one with weight 2)
- Each risk has an “**a-priori likelihood**” – its probability of occurrence if nothing is done to inhibit it
- Each mitigation has a “**cost**” – the resource cost (or costs, if multiple of these are being tracked in the study) of performing the mitigation, expressed in the appropriate cost units (e.g., $).

Finally, two kinds of relationships connect instances of the above:

- An **impact** relationship connects a risk to an objective, and has an associated value, a number in the range 0-1, indicating the proportion of that objective that would be lost were a risk to occur.
- An **effect** relationship connects a mitigation to a risk, and has an associated value, a number in the range 0-1, indicating the proportion by which that risk will be reduced were that mitigation to be applied (a further distinction among mitigations dictates whether the reduction will be to the risk’s likelihood, or to the risk’s impact(s)).

For the genesis of DDP see [Cornford, 1998]; for information about the tool and details of its treatment of risk, see [Feather and Cornford, 2003]; for a discussion of its application to studies of novel technologies, see [Feather et al, 2005].

3 Tool Integration and Semantic Dissonance

We describe the factors that motivated our attempts to integrate various pairings among the four tools described in the previous section. We then outline a number of instances of information clashes attributable to semantic dissonance that we identified as we prepared to integrate those tools. These clashes are impediments to integration. The section after this describes the solutions that we have employed to date, and how effective they have been, at surmounting these impediments.

3.1 Motivations for integration

On behalf of NASA, JPL designs, develops, operates and/or manages deep space missions, and spacecraft, instruments and/or other components of those missions. These activities span projects of widely varying scale (from individual instruments to multi-billion-dollar missions), and involvement may be from the early conceptual design phases all the way through to operation and decommissioning. This variety and range means that no single risk tool is suited to all applications, so it is not surprising to see a number of risk tools in use, different tools for different projects, and different tools at the different stages of a project's lifecycle. The four tools of our integration study are representative examples of this variety. RAP is used by the concurrent engineering team in the conceptual design phase, when mission concepts are explored. RAP is designed to be an agile, multi-user tool to suit this rapid-paced setting. DDP is typically employed to do a more in-depth study of a particular problem area, especially when there is need to scrutinize novel problems requiring judicious selection of the approaches to deal with them. FMEA is representative of the easy-to-adopt practices that help projects identify risk concerns. For example at the software/hardware interface, applications of fault trees, FMEAs and hazard analyses (and combinations of them, e.g., [Lutz & Woodhouse, 1997]) are commonly applied. PRA approaches are most appropriate for high-criticality systems and their detailed designs.

In this context we see many instances when it would be advantageous to have the capability to exchange information among these tools, and to use a combination of those tools' respective strengths. For example, in an earlier study [Cornford et al, 2003] in which Probabilistic Risk Assessment and DDP were separately applied to the same spacecraft design, 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, and 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. Relatively agile tools such as the FMEA tool (which we saw used to gather risk information about a software subsystem and its hardware interface) and RAP are in repeated use to gather and quickly assess risk information, but it would often be advantageous to be able to smoothly transition into using the more elaborate approaches: DDP when the mitigation options are plentiful and intertwined (meaning identifying a cost-effective approach to risk mitigation is a challenge), and PRA tools, such as Galileo, when there are non-trivial causal chains of faults leading to failures (e.g., as would be the case when redundant [backup] design elements are employed).

3.2 Information clashes from semantic dissonances between the tools

In contemplating building mechanisms for information exchange between these tools we identified the following information clashes, each an instance of semantic dissonance:

- Quantitative vs. qualitative assessments: the FMEA tool uses ordinal scale numbers to indicate likelihoods and severities of risks (values in the range 1-10, where a value of 1 is the least likely/severe, 10 the most), and effectiveness of detections, whereas the other tools use cardinal scales (or *ranges* of cardinal scale values – see next bullet) for these.
- Point values vs. ranges of values: the DDP uses point values to indicate likelihoods (e.g., 0.1), and severities of risks, whereas the RAP tool uses values for risks' likelihoods and severities

that correspond to several pre-determined *ranges* (e.g., a likelihood range of 0.7 – 1.0). Galileo can accommodate point values, and more generally, *distributions* of values (constant, exponential, lognormal, or Weibull).

- Risk mitigation described as pre- and post- risk conditions, vs. described as a transformation (e.g., "*halves* the likelihood"): RAP defines the effect of mitigations by the expressing the effected risks' post-mitigation likelihoods and severities (e.g., an unmitigated risk has a likelihood in the 0.7 – 1.0 range, and a mitigated likelihood in the 0.3 – 0.5 range), whereas DDP defines effects as a transformation on whatever was the pre-mitigation value (so if a mitigation has an effectiveness of 0.8 against a risk's likelihood, then applying the mitigation will decrease whatever likelihood that risk has by 80%, e.g., would reduce a likelihood of 0.7 by 0.56, to 0.14)
- Implicit vs. explicit representations of causality: Galileo's *fault trees* explicitly capture the causal structure of combinations of events leading to risks; the FMEA tool separately represents "cause" and "effect", but does represent any further detail of a causal structure beyond this single layer (although we have seen users indicate such structure in the textual descriptions – e.g., a system reboot caused by a "power surge or drop, or internal software error", but such textual descriptions are uninterpreted by the tool); similarly, both DDP and RAP have the concept of an Objective, instances of which are associated to risks to indicate how the occurrence of those risks would adversely affect those objectives – similar to the FMEA tool, these encode a single layer of cause and effect, but not a more elaborate structure; the RAP tool allows for the recording of "events" associated with risks, but does not utilize these in the calculation of risk likelihoods.
- Lack of corresponding concepts: DDP, RAP and the FMEA tool each have the explicit concept of a mitigation that reduces risk (albeit with rather different detail, see next bullet), yet Galileo does not (it evaluates a design as-is). Conversely, through its support of dynamic fault tree concepts, Galileo is able to much more faithfully represent the nuances of a given system design than any of the others.
- Differing levels of granularity (what is an atomic object to one tool may be a composite of several distinct objects to another): both RAP and the FMEA tool are organized to allow the association of a single "mitigation" (in RAP terms) / "current design control" (in FMEA terms) with a given risk, whereas DDP allows for multiple mitigations to be associated with a given risk. (Note that all three tools do allow a mitigation to be associated with multiple risks.)

4 Overcoming semantic dissonance impediments

We employed a mixture of solutions to the semantic dissonance impediments listed in the previous section. We present them in the subsections that follow, in order of their increasing complexity.

4.1 Transfer

Some information is readily transferred as-is (or nearly so) – e.g., risk titles and descriptions are textual fields common to all four tools.

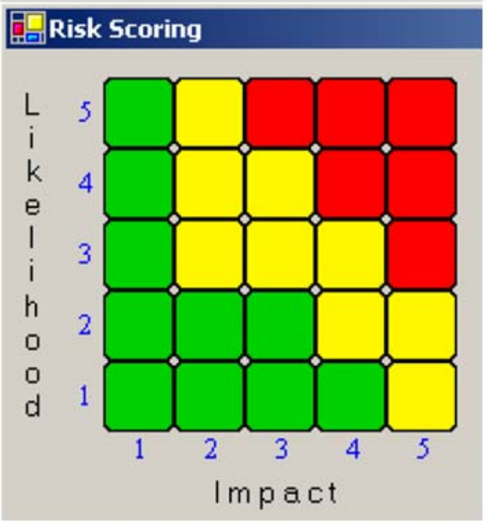
Adjustments may be needed for *syntactic* mismatches (e.g., when one tool's maximum length allowed for a title is greater than another's), and occasional *semantic assumptions* associated with names (e.g., tools such as Galileo assume that two risks with the same name are the exact same risk, whereas this is not necessarily so in RAP or DDP, wherein two different risks may happen to have been named the same). Simple translation mechanisms typically suffice to resolve these slight mismatches.

4.2 Translate

Some information from one tool can be translated into a representative equivalent in another tool.

An example of this is to translate a RAP score that is a *range* of values into a representative single point value in DDP. There can be a choice of such representative values. **Figure 0** shows RAP's interface for "scoring" risks: the user selects a cell in the 5x5 matrix, where the axes are scores are integers in the range 1-5. The correspondence between these scores and value ranges is shown in the table to the right, along with three possible representations of such ranges as point values. Translation in the reverse direction – from a point value to the range in which that point belongs – is of course unambiguous, but loses information (multiple points translate to the same range).

Translation can be used to convert *qualitative* data into a *quantitative* representation. For example we translated the FMEA tool's ordinal scale values (integers in the range 1-10) into DDP's cardinal scale values (floating point values in the range 0-1) by treating the ordinal as a regular number, and dividing by 10 (e.g., FMEA's lowest non-negligible likelihood, indicated by a "1", is translated into the probability 0.1 in DDP). Such a translation is obviously problematic – for example, the FMEA ordinals "1" and "2" in fact indicate only relative ordering, and are not a guarantee that the former is half the likelihood of the latter, whereas such an interpretation is made once these are translated into probabilities of 0.1 and 0.2. This is a recurring problem, and the best we can offer is to make the translation explicit (and adjustable). We have struggled with this same problem in earlier work that merged DDP with "Risk Balance Profiles" – the latter being DDP-like information, but with only qualitative values – see [Feather and Cornford, 2003] for more discussion of this issue. We note an ingenious treatment of this problem described in [Chiang & Menzies, 2002]: a qualitative scale is translated into a quantitative scale that preserves the ordering, but with randomly chosen delineations; the ingenuity is to do this repeatedly (with the delineations chosen randomly again each time), perform the qualitative reasoning on each result to derive some number of qualitatively-based insights, and look to see which of those insights are relatively "stable" (occur repeatedly, except perhaps for some of the extremes).



Risk Scoring

Likelihood					
5	🟩	🟨	🟥	🟥	🟥
4	🟩	🟨	🟨	🟥	🟥
3	🟩	🟨	🟨	🟨	🟥
2	🟩	🟩	🟩	🟨	🟨
1	🟩	🟩	🟩	🟩	🟨
	1	2	3	4	5
	Impact				

RAP score	Range	Alternate Representative Values		
		Max	Mean	Min
5	1.0 – 0.7	1.0	0.85	0.7
4	0.7 – 0.5	0.7	0.6	0.5
3	0.5 – 0.3	0.5	0.4	0.3
2	0.3 – 0.01	0.3	0.145	0.01
1	0.01 – 0.0	0.01	0.005	0.0

Figure 1 – RAP 5x5 Risk Scoring, and the corresponding range values

4.3 Extend

Extend a tool with a concept it lacks (taking that concept from another of the tools).

Our primary example of this is the addition to DDP of the concept of a *range* of values (as seen in RAP). DDP initially dealt with a single point value for a risk's likelihood (and similar point values for a risk's impact on an objective, and for a mitigation's effect at reducing that risk). We extended DDP to hold for each of these a *triple* of values, one the "minimum" value, one the "maximum", and one the "nominal". By default DDP presents and manipulates just the "nominal" value, but offers the option of turning on the capability to use all three. This allowed us to represent RAP data, translating the end points of a RAP value range into the minimum and maximum values (and continuing our earlier practice of approximating the range with a single nominal value, the mean of the two).

Extensions such as this are on the whole conceptually straightforward, but nevertheless can have extensive ripple effects through the existing capabilities of the tool into which they are introduced. For example, DDP's various visualization capabilities (e.g., bar charts that plot risk levels, objective attainment levels, etc, 2-D plots of risks plotted against axes of likelihood and consequence) had been constructed with the assumption of single values. To date we have extended just the bar chart visualizations to indicate ranges of values (and to allow sorting the bars with respect to either of the extremes). See Figure 1, showing DDP's bar chart plots of (a) nominal values, (b) extremes superimposed as red (upper) and black (lower) line segments (note that the plot is a log scale, so some of the lower extremes do not show up), and (c) an alternate display indicating just the extremes.

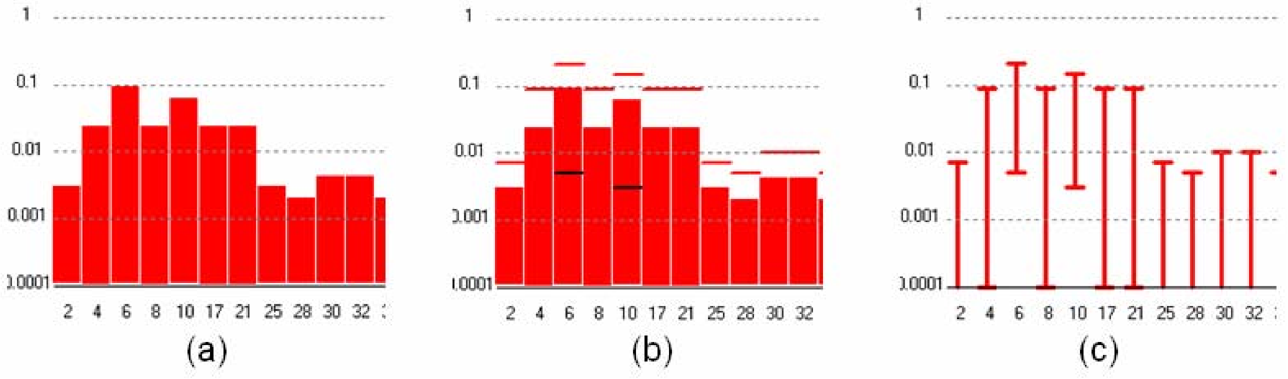


Figure 2 - DDP's bar chart plot extended to show ranges of values

4.4 Restrict

A complementary approach to extending one of a pair of tools is to restrict the other! This means constraining usage of the tools' features (perhaps leaving this to the users to be good citizens, or perhaps enforcing this via a mechanism within the tool itself) to the subset that matches another tool's capabilities.

An example follows from our extension of DDP to implement ranges of values. The DDP implementation is more general than RAP's: in RAP there is a pre-determined set of 5 ranges (1.0 – 0.7, 0.7 – 0.5, etc); In DDP we have not restricted the possible pairs of maximum and minimum values, so that it is possible, through the DDP interface, to assign a range that is not one of RAP's 5 ranges. For example, the range 1.0 – 0.6 would span RAP's topmost range and extend partway into its next one. Were we to restrict use in DDP only the pairs of extremes that match RAP's ranges, then we would continue to guarantee the ability to transfer such information as-is back to RAP.

4.5 Meld

By "melding" we mean a more semantically intensive version of extension, in which the incorporation of one tool's concept into another tool requires some in-depth considerations of the semantic implications (it's *not* conceptually straightforward).

Our primary example of this is the melding of fault trees (from Galileo especially, but also sometimes implied by what we saw in FMEA uses) into DDP. The semantic complications stem from the interplay between fault trees and mitigations. Figure 0 shows (a) DDP's standard topology for how Objectives, Risks and Mitigations are linked, and (b) where fault trees fit into this.

In standard DDP, where Risks are atomic objects, an "Effect" link between Mitigation and Risk is accompanied by a value, the proportion by which that Risk is reduced if the Mitigation is applied. The nature of the Mitigation dictates whether the reduction is to the Risk's severity (in DDP, its Impact(s) on Objectives) or to the Risk's likelihood; the latter has a further distinction between "preventative" measures that decrease the likelihood of the risk occurring in the first place (e.g., adopting a coding standard to help avoid naming confusions among a team of programmers), and "detection" measures that decrease the likelihood of a risk present at one phase from going undiscovered (e.g., software testing reveals bugs that are then repaired, so that the final product is less "buggy"). When fault tree structures replace DDP's atomic risks, some non-trivial semantic questions arise, e.g., does it make sense for a Mitigation that reduces a Risk's severity to be connected to *any* node within a fault tree? (the answer is no; it only makes sense to connect such Mitigations to fault tree nodes that are connected, via "Impact" links, to Objectives – generally speaking the "root" nodes of fault trees, but sometimes it makes semantic sense for an intermediate node can have such Impact links.) For a more thorough discussion of the details of this, see [Feather, 2004]. Suffice it to say that this is an instance of integration that raises some interesting semantic issues, significantly beyond data value conversion issues.

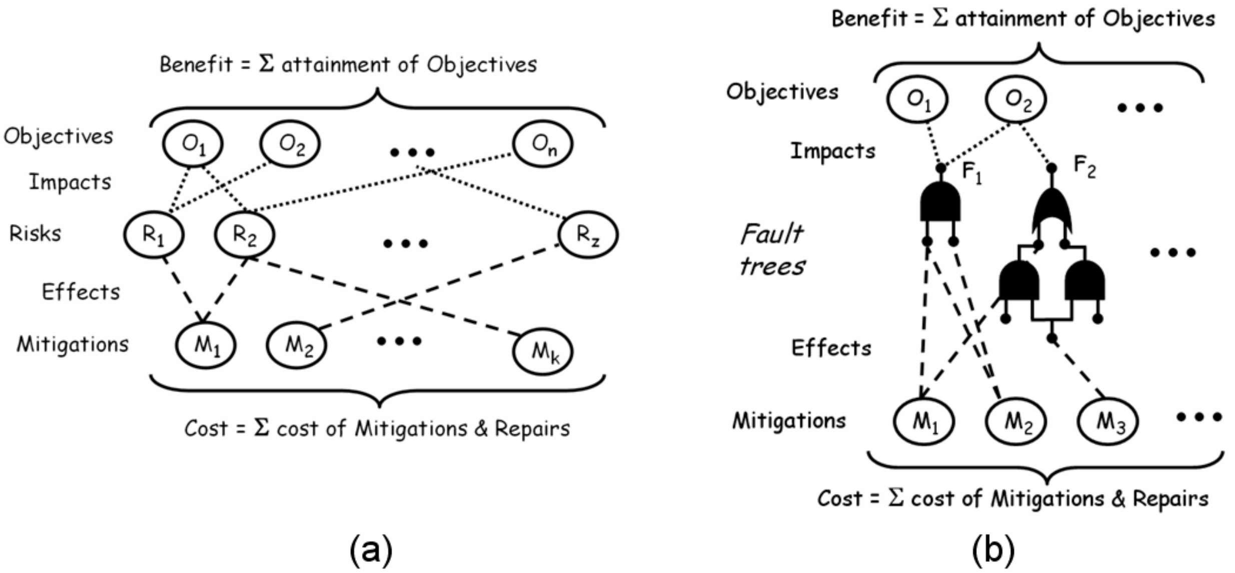


Figure 3 - blending fault trees into DDP's structure of Objectives, Risks and Mitigations

4.6 Alternate

The different capabilities of a pair of risk tools can be used in alternation without necessarily requiring each tool to understand the full range of semantics of the other. This is done by exchanging information that only one tool manipulates – the other tool treats it as opaque data, akin to the way that browsers treat "cookies" (this usage derives from the UNIX term "magic cookie").

Two examples of this are seen in our combination of Galileo and DDP extended with "and" and "or" fault trees:

- In extending DDP to handle such trees, we extended the DDP code to calculate the likelihoods of such trees. However, DDP's implementation is inefficient for fault trees that contain large numbers of shared events. In contrast, Galileo is much more efficient – it makes use of binary decision diagrams (BDDs) to solve static sub-trees [Doyle & Dugan, 1995]. To take advantage of this, we added a mode to DDP in which it calls upon Galileo to perform such evaluations. DDP passes Galileo a fault tree whose likelihoods for the leaf nodes have been determined within DDP (taking into account the likelihood-reducing effects of DDP “mitigations”), and Galileo computes and returns to DDP the likelihood of the root of that tree.

- Another use of alternation in this same realm would be to handle the other kinds of fault tree gates that Galileo supports. Recreating the semantics of these within DDP would be a significant effort. Instead, we could use this “alternate” strategy to have DDP store the representation of such gates, but again rely upon Galileo to evaluate the likelihoods of trees that make use of them. Note – we haven’t implemented this yet, but believe it to be feasible.

5 Utility

The introduction section described various motivations for integration of risk tools. Here we illustrate some of the utility gained from the integrations among the four risk tools we studied: DDP, the FMEA spreadsheet, Galileo and RAP.

5.1 FMEA & DDP

Our integration of the FMEA spreadsheet and DDP allowed for DDP’s computations and visualizations to be brought to bear on the data gathered using the (more standard and easy to use) FMEA spreadsheet. The correspondences we established between the FMEA information and its DDP representation were as follows:

FMEA concept	**DDP concept**
Item/Function’s potential Failure Mode	Fault Tree root, an “or” node combination of its causes
Potential effect of failure	Objective to avoid such an effect
Severity	Impact of fault tree root on Objective
Potential cause(s)/mechanism(s) of failure	Fault tree leaf(leaves)
Probability	Likelihood of fault tree leaf
Current Design Control	Mitigation
Detectability	Effect of Mitigation on reducing Likelihood or Severity (note: we rely upon a human to disambiguate among these)

To achieve this we made key use of the *meld* of fault trees within DDP; this then allowed us to *transfer* various FMEA concepts into DDP equivalents, and *translate* FMEA’s qualitative values into DDP’s quantitative values.

The original spreadsheet form made *some* risk questions easy to answer, for example “which failure mode contributes the most risk?” – the spreadsheet is set up to calculate the “Risk Priority Number”, the produce of a failure mode’s severity, probability and (lack of) detectability, so higher numbers indicate more risk. Once within DDP, it became possible to answer *additional* risk

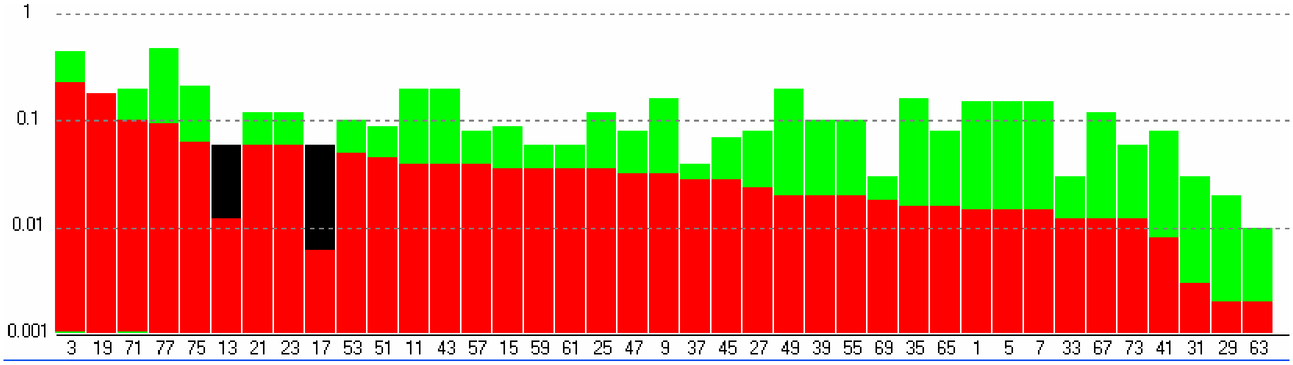


Figure 4 - DDP visualization of the FMEA data, showing a “what-if” scenario

questions, for example “which current Design Control reduces the most risk?”, and explore “what-if” scenarios, for example, “how would risks change if Design Control <x> were not done?”. DDP can both calculate the above, and present the results via its cogent visualizations. Figure 0 shows an example of such based on data from one of our project’s uses of the FMEA spreadsheet: each vertical bar denotes a failure mode; red (green) heights = total risk taking (not taking) the reducing effects of current design controls into account; black height = increase in risks from the “what if” of turning off one of the current desing controls (with the bars sorted in descending order of mitigated risk). From this kind of visualization it is easy to see the “big picture” of how risks compare to one another, where the current design controls are having their effects (e.g., the second-tallest risk does not have a green portion to its bar, indicating that none of the current design controls have a mitigating effect on that risk), and what would be the effect of changes (e.g., the black portions of two of the bars indicate that turning off that one current design control would raise both those risks, turning them from well-mitigated to the 6th and 9th positions in risk order).

5.2 RAP and DDP

Our integration of RAP and DDP primarily allowed for scrutiny of RAP-collected risk data using DDP’s visualizations. In a case study we used this to identify the vulnerabilities of the system being studied, and thereafter used DDP as a stepping-stone to the more detailed analysis of the vulnerable parts of the design using Galileo (see next subsection for the DDP-Galileo connection). The RAP-DDP correspondences we established were as follows:

RAP concept	**DDP concept**
Objective	Objective
Team members (e.g., “Power”, “Propulsion”)	Risk category values
Risk element (possibly shared by multiple of the team members, e.g., the same risk might have different impacts as viewed by the Power team member and the Propulsion team member)	Risk folder (a grouping of individual risks)
Risk factor	Risk
Risk factor’s owner	Risk category
Risk likelihood and impact on objective(s) *before* mitigation	Risk’s unmitigated likelihood and impact on objectives
Risk likelihood and impact *after* mitigation	Risk-reducing effect of that mitigation on that risk

To achieve this we used *translate* to turn RAP team members into the possible DDP risks’ “category” values (string values in an enumerated set), *transfer* to pass over the RAP objectives titles, and for risks, their titles, and (pre-mitigation) likelihoods and impacts on objectives. We had a choice of whether to *translate* a RAP value range into a representative single point value in DDP, or to *extend* DDP to represent ranges (we explored both alternatives). Finally, we were able to *translate* the RAP style of expressing a mitigation’s effect on a risk via its post-mitigation likelihoods and severities into the DDP equivalent of mitigation(s) that translate (e.g., “halve”) risk likelihoods or impacts. This last proved the most complex translation, since it commingled solutions to several semantic dissonance obstacles, as follows:

- In RAP, the post-mitigation risk can differ from its pre-mitigation state by changes to both likelihood and impact. The assumption in DDP is that a mitigation changes only one of these. A possible solution would have been to extend DDP’s mitigations to liberalize this assumption; however we instead took the route of translation into a pair of coupled DDP mitigations, one of which effected likelihood, the other impact.
- If translating a RAP value range to a representative single point value in DDP, then:
  pick the translation to use (e.g., the “Max” column of the table in Figure 0);
    let P = the representative single point value for the RAP pre-mitigation score, and

let Q = the representative single point value for the RAP post-mitigation score;
recall that a DDP mitigation effect value V indicates the reduction proportion, so a V-valued mitigation applied to P reduces it by P*V, i.e., Q = P * (1 – V)
So V = 1 – Q/P

For example:

RAP pre-mitigation range 1.0 – 0.7 becomes representative single point value 1.0 (P)
RAP post-mitigation range 0.3 – 0.01 becomes single point value 0.3 (Q)
DDP mitigation effect value V is 1 – Q/P = 1 – 0.3/1.0 = 0.7

- If using DDP is extended with *ranges* of values, then translating mitigation of a RAP range into DDP is accomplished as follows:

  let P-pessimistic = the high end of the pre-mitigation RAP range
  let P-optimistic = the low end of the pre-mitigation RAP range
  let Q-pessimistic = the high end of the post-mitigation RAP range
  let Q-optimistic = the low end of the post-mitigation RAP range
  Use these to calculate a DDP mitigation effect with a value range V- pessimistic – V-optimistic, as follows:
  V-pessimistic = 1 – Q-pessimistic/P-pessimistic
  V-optimistic = 1 – Q-optimistic/P-optimistic

  See Figure 1 for an example.

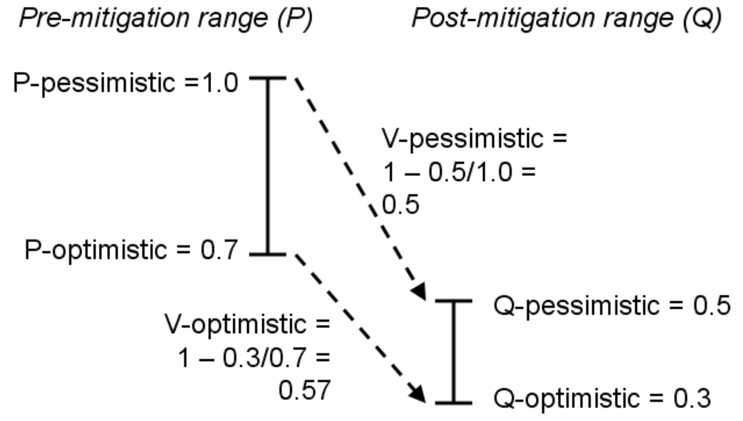


Figure 5 - translation of a range-valued pre- and post- mitigation risk

The net effect of all this is that we are able to use DDP to visualize RAP-collected risk information.

An example of such is seen in **Error! Reference source not found.**, where data from a RAP study has been translated into DDP extended with ranges of values. The upper half of the figure shows the entire set of risk information, where the possibly several risks factors of a risk element are grouped adjacently. The “zoom” to the left (which isn’t a part of DDP – we added it to form this figure) shows a group of a risk factor’s four such risk factors – the different colors indicate the different team members assessments of that risk element. The lower half of the figure shows the same set of risk information, but sorted in decreasing order of each risk factor’s maximum risk instead of grouped by risk element.

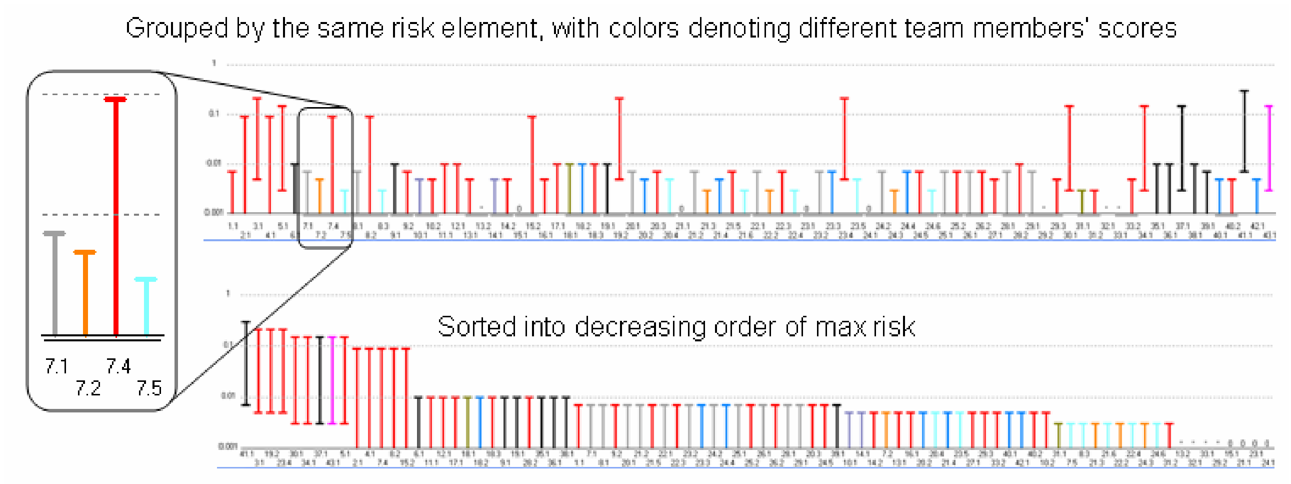


Figure 6 - DDP visualizations of RAP risk data

5.3 DDP and Galileo

Our integration of DDP and Galileo allowed for the combination of DDP's notions of mitigations as options for reducing event likelihoods and risk severities, and Galileo's fault tree notions of calculating the likelihoods of fault tree root nodes from the structure of those trees and their leaf node likelihoods. The correspondences we established between DDP and Galileo were as follows:

Galileo concept	DDP concept
Fault tree of "and" and "or" gates	Fault tree of "and" and "or" gates
Fault tree's impact(s)	Fault tree's impact on Objective(s)
Fault tree's leaf node likelihoods	Fault tree's leaf node likelihoods

The precursor to this was the *meld* of fault trees into DDP. As discussed earlier, this required some in-depth considerations of the semantic implications – see [Feather, 2004] for details. Having incorporated fault tree notions into DDP, we *transfer* DDP fault trees to Galileo, and have the latter compute their root node likelihoods.

An example of this is seen in tbd, a study we performed on a representative example of spacecraft design information. The upper half of the figure shows the DDP structure of Objectives (represented by the top row of blue circles) linked to root nodes of the Fault Trees (in the middle); the occurrence of the fault tree root nodes would detract from the attainment of Objectives to which they are linked. Linked to the Fault Tree leaf nodes are the Mitigations (represented by the bottom row of green circles) that would decrease the likelihoods of those leaf nodes. We automated the transfer of each of the DDP fault trees into their equivalent within Galileo. This is illustrated by the boxed fault tree in the DDP structure, and its Galileo equivalent shown in the lower half of the figure. The net result of this is that we employ Galileo's capabilities for evaluating fault trees, and DDP's capabilities for considering alternate selections of Mitigations. For a given selection of Mitigations, DDP computes its *cost* and *benefit*. Cost is simply the sum of the costs of the selected Mitigations; benefit is the sum of expected attainment of the Objectives, taking into account the risks (fault tree roots) that detract from those Objectives. The likelihoods of those fault tree roots are used in this calculation; they are calculated by Galileo from knowledge of the fault tree structures and their leaf node likelihoods. These last are set by DDP, taking into account the likelihood-reducing effects of the selected Mitigations.

The net result of this is that we can evaluate a selection of Mitigations in terms of its cost and its benefit. In this study, there were 88 different Mitigations, hence 2^{88} ways of selecting from among them. To explore this size of search space we use simulated annealing (a standard form of heuristic search that is programmed within DDP) to locate near-optimal selections of mitigations. The plot of

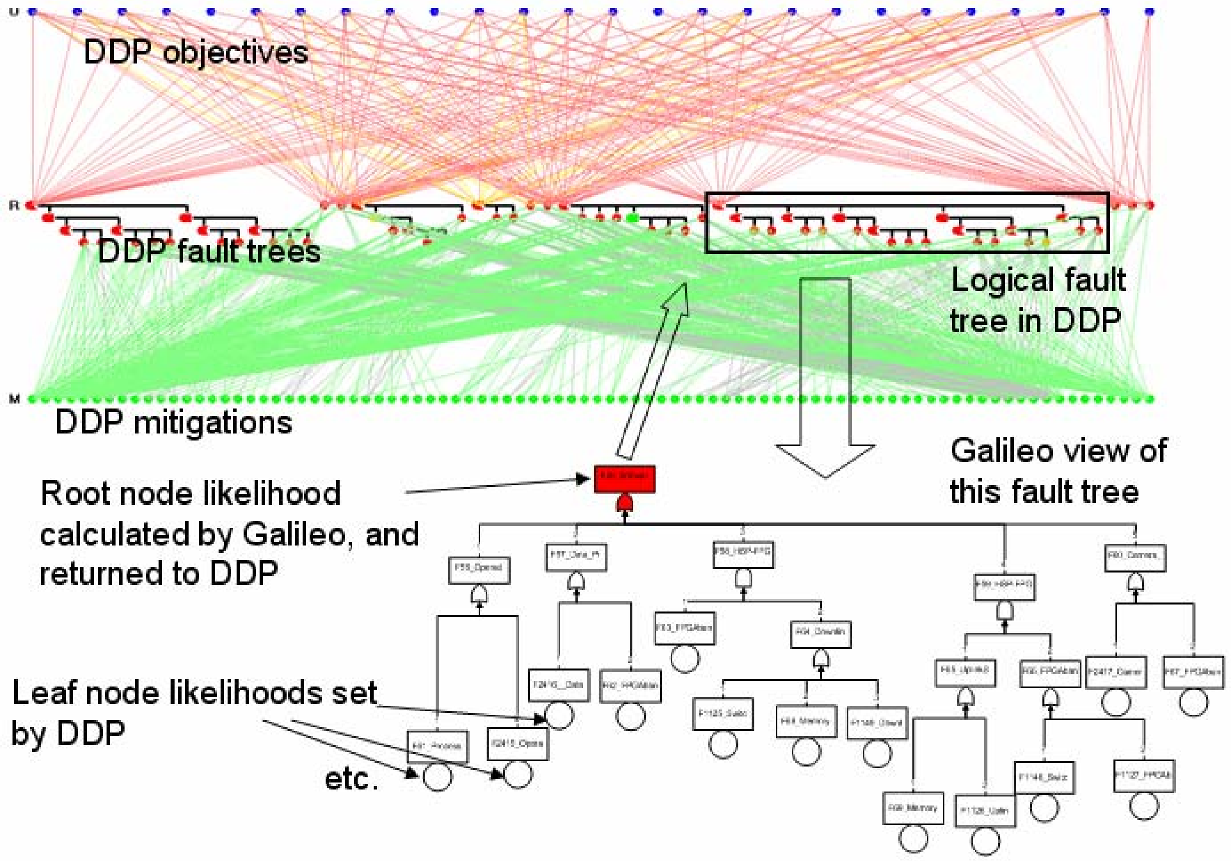

U
DDP objectives
R
DDP fault trees
Logical fault tree in DDP
M
DDP mitigations
Galileo view of this fault tree
Root node likelihood calculated by Galileo, and returned to DDP
Leaf node likelihoods set by DDP
etc.

Figure 7 - exchange of fault tree information between DDP and Galileo

a search run where we utilized the DDP-Galileo integration to search this design space is shown in tbd. Each tiny red point on this plot represents a different selection of Mitigations, for which the DDP-Galileo integration has been called upon to evaluate its cost and benefit. For the cost bound we set for this run (indicated by the vertical green line), the run found a wide spread of selections; those to the left of the green line and as high up as possible are the optimal ones. Overall this shows the considerable power that can be gained from the integration of sophisticated capabilities that different tools offer.

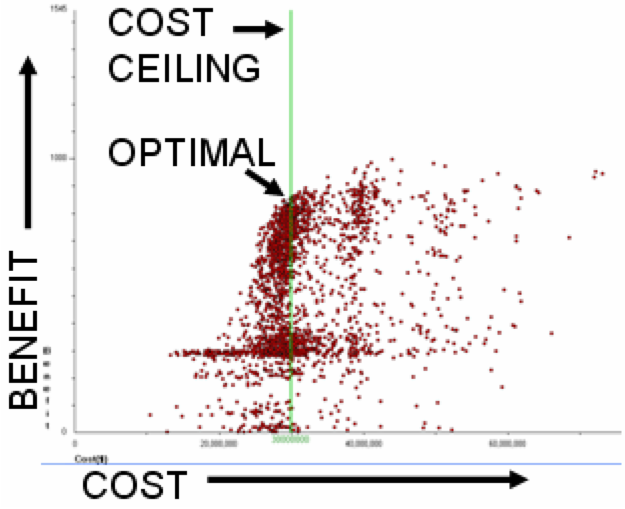


Figure 8 - plot of a search for near-optimal selection of Mitigations

6 Discussion and Future Work

In this paper we have described our motivation for seeking to integrate several risk tools, and our experiences in doing so. The focus has been on the issue of semantic dissonance – mismatch between conceptual assumptions made by separately developed tools. We have described the kinds of mismatches that we encountered, our solutions to overcoming them, and the utility we gained from the integrations that then became possible. Figure 9 gives an overview of our work – the main features and applications of each of the tools are listed, their major outcomes are shown at the top and bottom, and arrows between the tools indicate the information transfers among them that we have utilized. The FMEA-to-Galileo connection is one that we have not yet implemented, but believe can be accomplished with a subset of the techniques we developed for the FMEA-to-DDP integration.

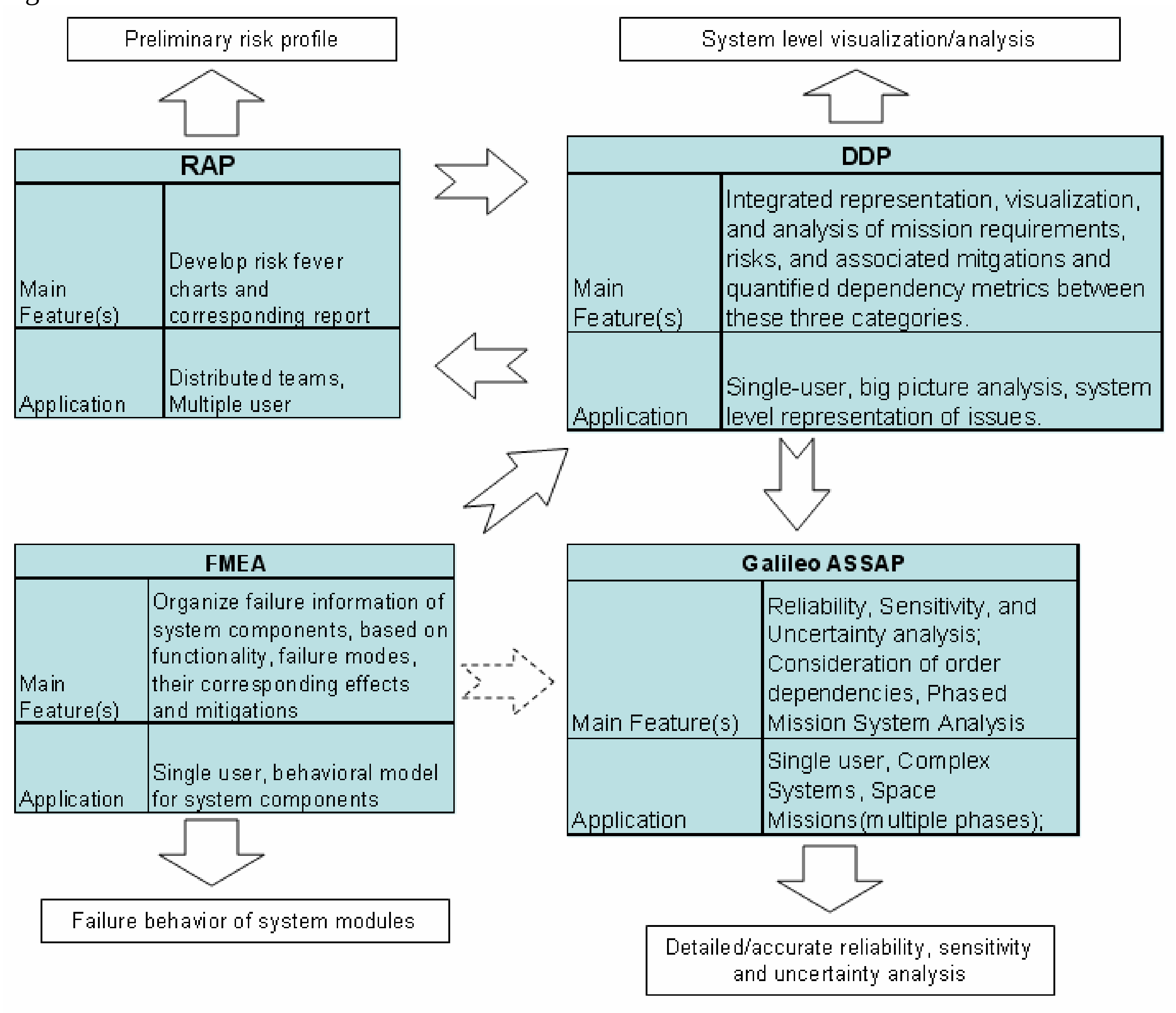

Preliminary risk profile
System level visualization/analysis
RAP
Main Feature(s)
Develop risk fever charts and corresponding report
Application
Distributed teams, Multiple user
DDP
Main Feature(s)
Integrated representation, visualization, and analysis of mission requirements, risks, and associated mitgations and quantified dependency metrics between these three categories.
Application
Single-user, big picture analysis, system level representation of issues.
FMEA
Main Feature(s)
Organize failure information of system components, based on functionality, failure modes, their corresponding effects and mitigations
Application
Single user, behavioral model for system components
Galileo ASSAP
Main Feature(s)
Reliability, Sensitivity, and Uncertainty analysis; Consideration of order dependencies, Phased Mission System Analysis
Application
Single user, Complex Systems, Space Missions(multiple phases);
Failure behavior of system modules
Detailed/accurate reliability, sensitivity and uncertainty analysis

Figure 9 - overview of our use of information transfers made possible by our integrations

To date, our most significant use of these combined capabilities has been to start with risk data generated from a study conducted at JPL's Project Design Center, using the RAP tool; this information was then transferred to DDP to help identify the key vulnerabilities of the study design;

this information was then transferred in turn to Galileo, to perform detailed analyses of those vulnerable parts. For details on this, see [Meshkat et al, 2005].

We plan to study the integration of additional risk tools in a similar manner. Two in particular that we are currently working with are:

- Active Risk Manager [ARM] – this is a commercial risk management software system in use at NASA. At its heart, its representation of risks and mitigations is akin to that of the RAP tool (albeit with many more attributes that support project management activities, and with more options for how risks are scored). We anticipate that the approach we followed to connect RAP and DDP will serve as the basis for connecting ARM and DDP.
- "Event Consequence Tree" (ECTree) – a NASA-developed spreadsheet-based tool for constructing and calculating probabilities of system failures. At its heart is the commonly used concept of an event tree (ET) – described in [NASA PRA, 2002] as "An ET starts with the initiating event and progresses through the scenario, a series of successes or failures of intermediate events called pivotal events, until an end state is reached." Likelihoods are associated with those pivotal events. Our preliminary experiments at integrating DDP and ECTree are following what we called in Section 4.6 the *alternate* approach to their integration – DDP controls the likelihoods of the pivotal events, and ECTree takes those into account to compute the overall likelihoods of the different end states, passing those likelihoods back to DDP.

The starting motivation for our work was the ambitious vision of a "Risk Tool Suite for Advanced Design" (RTSAD). RTSAD was to have included not only risk tools, but also risk data repositories, all woven into a support system for designers. At the heart of the RTSAD architecture was to be a unified representation of risk information, which would serve as the nexus through which all risk tools would communicate. In contrast, our work to date has concentrated on the pairwise integration of risk tools. Although the RTSAD effort is not continuing, our hope is that we will learn from our studies what needs to go into a unified risk representation, and so be of assistance to future visionary risk integration efforts. Our results so far point to a middle ground of complexity in tool integration. In the risk tools we studied, there appears to be sufficient "semantic overlap" between the different tools to make their integration more complex than could be readily understood as ontology matching (e.g., see [Doan et al, 2004]). Conversely, the tools are sufficiently different there has not yet been developed for them a single unifying semantic domain (e.g., in the realm of reliability modeling and analysis, [Coppit et al] use failure automata to formalize dynamic fault trees and reliability block diagrams). We believe this middle ground is worthy of continued attention, so as to achieve the gains to be had from the integration of disparate risk tools.

7 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 and funded through the internal Research and Technology Development program, NASA's Office of Safety and Mission Assurance, and the former NASA Codes R and T. 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.

We especially thank Chester Borden, Yuri Gawdiak, Kenneth Hicks, Jelly Moran, Irem Tumer & Stephen Prusha for making this work possible, and Jane Malin & David Throop for interesting discussions on issues of risk integration

8 References

[ARM] Strategic Thought Limited, "Active Risk Manager", http://www.strategicthought.com

[Chiang & Menzies, 2002] E. Chiang & T. Menzies, "Simulations for Very Early Lifecycle Quality Evaluations", *Software Process Improvement and Practice* 7(3-4): 141-159, 2002.

[Coppit et al, 2003] D. Coppit, R.P. Painter & K.J. Sullivan, "Shared Semantic Domains for Computational Reliability Engineering", *14th International Symposium on Software Reliability Engineering*, pp. 169-180, 2003.

[Coppit & Sullivan, 2003] D. Coppit & K.J. Sullivan, "Sound methods and effective tools for engineering modeling and analysis", 29th International Conference on Software Engineering, 3-10 May 2003, pp. 198-207.

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

[Cornford et al, 2003] S.L. Cornford, T. Paulos, L. Meshkat & M.S. Feather. "Towards More Accurate Life Cycle Risk Management Through Integration of DDP and PRA", *IEEE Aerospace Conference*, Big Sky, Montana, Mar 2003, pp. 2.1106 – 2.1200.

[Doan et al, 2004] A.Doan, J. Madhavan, P. Domingos & A. Halevy, "Ontology Matching: A Machine Learning Approach", in *Handbook on Ontologies in Information Systems*, pp. 385-403, S. Staab & R. Studer (eds.), Springer, 2004.

[Doyle & Dugan, 1995] S.A. Doyle & J.B. Dugan, "Dependability Assessment using Binary Decision Diagrams (BDDs)", *25th Annual International Symposium on Fault-Tolerant Computing*, Pasadena, California, 27-30 June 1995.

[Dugan et al, 1992] J.B. Dugan, S.J. Bavuso & M. Boyd, "Dynamic Fault Tree Models for Fault Tolerant Computer Systems," *IEEE Trans. on Reliability*, 41(3), Pages 363-377, Sept. 1992.

[Dugan et al, 1999] J.B. Dugan, K.J. Sullivan, and D. Coppit. "Developing a low-cost, high-quality software tool for dynamic fault tree analysis," *IEEE Trans. on Reliability*, Dec. 1999, pp. 49-59.

[Feather and Cornford, 2003] M.S. Feather & S.L. Cornford, "Quantitative risk-based requirements reasoning", *Requirements Engineering* (Springer), 8(4), 2003 pp. 248-265 – published online 25 February 2003, DOI 10.1007/s00766-002-0160-y.

[Feather, 2004] M.S. Feather, "Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface", *ISSRE 2004 - 15th IEEE International Symposium on Software Reliability Engineering*; Saint-Malo, Bretagne, France, Nov 2-5 2004.

[Feather et al, 2005] M.S. Feather, S.L. Cornford, K.A. Hicks & K.R. Johnson, "Applications of tool support for risk-informed requirements reasoning," *Computer Systems Science and Engineering* (CRL Publishing Ltd); 20(1): 5-17, January 2005

[Kent, 1978] W. Kent. "*Data and Reality: Basic Assumptions in Data Processing Reconsidered*". Elsevier Science Inc., 1978. ISBN 0444851879.

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

[Meshkat et al., 2003] L. Meshkat, S. Cornford, & T. Moran "Risk Based Decision Tool for Space Exploration Missions". *Proceedings of the AIAA Space Conference*, September 2003

[Meshkat & Oberto, 2004] L. Meshkat & R.E. Oberto, "Towards a Systems Approach for Risk Considerations during Concurrent Design", *United Nations Space Conference, Beijing, China*, May 2004.

[Meshkat et al, 2005] L. Meshkat, S. Cornford, L. Voss & M. Feather "An Integrated Approach to Risk Assessment for Concurrent Design", *IEEE Aerospace Conference*, Big Sky, Montana, March 2005.

[NASA PRA, 2002] 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; http://www.hq.nasa.gov/office/codeq/doctree/praguide.pdf

[RIM] HL7 Reference Information Model (http://www.hl7.org/library/data-model/RIM/modelpage_non.htm)

[STEP] Michael J. Pratt, "Technical Note: Introduction to ISO 10303 – the STEP Standard for Product Data Exchange", http://www.mel.nist.gov/msidlibrary/doc/jcise1.pdf

[Sullivan et al, 1999] K.J. Sullivan, J.B. Dugan & D. Coppit, "The Galileo fault tree analysis tool", *Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*, 15-18 June 1999 pp. 232-235.

[Throop et al, 2005] D.R. Throop, J.T. Malin & L. Fleming, "Knowledge Representation Standards and Interchange Formats for Causal Graphs", *IEEE Aerospace Conference*, Big Sky, Montana, March 2005.

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