Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation

Martin S. Feather* Steven L. Cornford* James D. Kiper** Tim Menzies***
* *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA*
** *Miami University, Oxford, OH*
*** *West Virginia University, Morgantown, WV*
{Martin.S.Feather, Steven.L.Cornford}@jpl.nasa.gov, kiperjd@muohio.edu, tim@menzies.us

Abstract

For several years we have been employing a risk-based decision process to guide development and application of advanced technologies, and for research and technology portfolio planning. The process is supported by custom software, in which visualization plays an important role. During requirements gathering, visualization is used to help scrutinize the status (completeness, extent) of the information. During decision making based on the gathered information, visualization is used to help decision-makers understand the space of options and their consequences.

In this paper we summarize the visualization capabilities that we have employed, indicating when and how they have proven useful.

1. Introduction

The purpose of this paper is to report on our experiences using visualization to help in the early phases of project planning. This is the time when requirements are being determined, and planning is done for the entire development to follow. Key decisions are made, including: determination of purpose, from which stem detailed requirements; planning the rest of a system's lifecycle, including subsequent development, testing, deployment, maintenance and future upgrades; allocation of resources to those phases (e.g., budget, schedule, testing platforms), and the determination of the architecture and early phase design.

Our experiences suggest that there is a non-trivial amount of information available even during the earliest phases of the lifecycle. This information comes from many sources ("stakeholders" in the parlance of the software community) – the customers, funders, managers, developers, users, etc. Information may take the form of guidance extracted from past experience, coupled with experts' best estimates in the face of novel aspects of the effort. Novelty stems from new applications, new circumstances, new hardware and software resources, new development methodologies, etc.

Decision making during these early phases remains a predominantly human activity, but one which can and should be well-informed by this wealth of available information. Visualization can play a prominent role by portraying the information in ways that make it amenable to extracting key insights, and open to scrutiny so that the information basis for the decision making is clear.

This paper illustrates our work in this area, based on a novel but effective approach to employing quantitative reasoning during the early phases of system (software and/or hardware) developments, and technology planning. We have previously published on various aspects of this work, references to which appear within the paper. Here we focus on how use of visualization plays a crucial role throughout.

2. Basis for our work

2.1. Fundamental concerns of early-lifecycle decision making

Our work takes place in the context of early-lifecycle planning of complex system developments. In particular, our focus is on spacecraft and spacecraft technologies, both software and hardware, in their early phases of development. Although this may seem an esoteric application area, the concerns that dominate early-lifecycle decision – cross disciplinary effects, resource constraints, need for reliability, and novel aspects of the problem – are common to many domains. In more detail, these concerns are as follows:

First International Workshop on Requirements Engineering Visualization (REV'06)
0-7695-2711-6/06 $20.00 © 2006 IEEE

- *Cross-disciplinary concerns* (e.g., navigation, propulsion, telecommunications). These concerns are cross-coupled and interact in multiple ways (e.g., electromagnetic interference, heat transfer).
- *Severe constraints* on the systems being developed and on the development process itself. Time and budget pressures constrain development; operational resources constrain the resulting system (e.g., mass, volume, power).
- *Reliability issues*. Spacecraft are critical systems that must operate correctly the first time in only partially understood environments, with no chance for repair.
- *Unknowns*: past experience provides only a partial guide when new mission concepts are to be enhanced and enabled by new technologies of which past experience is lacking.

Because of this combination of challenging aspects, invariably no one person has expertise that spans all the disciplines, or can simultaneously juggle all the factors involved in large and complex designs. Furthermore, much of the design skill is tacit knowledge in the heads of experts, so it cannot be pre-encoded in an automated tool. Our response has been to follow a process in which we convene experts, elicit the relevant knowledge from them, capture it in a simple but agile model, and use this amalgamated information to help those experts make decisions. We describe this process next.

2.2. A risk-based approach to assist requirements-time decision-making

Use of cost-benefit models for prioritizing software requirements was convincingly advocated in Karlsson and Ryan [14]. In their original concept, each requirement was scored in two dimensions – how much benefit that requirement would convey, and how much it would cost to implement. In our setting we have found that requirements are highly intertwined with the designs that can be contemplated to (partially) satisfy them. We therefore use a model from which cost and benefit can be *derived*. The model is then used to guide decision-making, e.g., to select from among alternative designs.

Our model uses a concept of Risks to interpose between requirements and designs, the latter being composed of choices among alternative ways to quell risks. This is somewhat reminiscent of van Lamsweerde & Letier's notion of Obstacles in Goal-Oriented Requirements [22], however, our model uses more quantification and less logical substructure. Our model is called Defect Detection and Prevention (DDP), the name reflecting its origins as a method intended for quality assurance planning of hardware systems [5]. In more detail, a DDP model is populated by instances of three kinds of concepts: *Requirements* – what it is that the system or technology is to achieve, *Risks* – what could occur to impede the attainment of the Requirements, and *Mitigations* – what could be done to reduce the likelihood and/or impact of Risks[1]. In the DDP model these instances have quantitative attributes: each Requirement has a *weight*, its relative importance; each Risk has a *likelihood*, its probability of occurrence, and each Mitigation has a *cost*, the cost of performing it – usually a financial cost, but other resources can also be considered, such as schedule, power, mass. Quantitative relationships connect these instances: Requirements are related to Risks, and Risks are related to Mitigations. Specifically, Requirements are related to Risks to indicate how much each Risk, should it occur, *impacts* (i.e., detracts from the attainment of) each Requirement. Risks are quantitatively related to Mitigations, to indicate how much of a Risk-reducing *effect* a Mitigation, should it be applied, has on reducing each Risk, either by decreasing the Risk's likelihood, or by reducing the magnitude of the Risk's impacts on Requirements; the nature of the Mitigation dictates which kind of reduction takes place.

The majority of DDP applications to date have been in the area of technology infusion [9]. DDP has proven helpful to clarify the definition of the mission requirements that the candidate technology will satisfy, and to identify and address early on the technology-specific engineering difficulties that may result from alternative technology/mission architecture decisions.

2.4. DDP models

A DDP model typically consists of dozens of instances of each of the three concepts (Requirements, Risks and Mitigations), and hundreds of linkages among them – Impacts link between Requirements and Risks, and Effects link between Risks and Mitigations.

This quantity of information, its convoluted nature, and its origins in the various discipline areas of spacecraft development, combine to make decision-making challenging. Especially problematic is the selection of Mitigations. In almost all applications the total cost of all the identified Mitigations far exceeds the resources available, necessitating the careful selection of which of them to perform. We find that

[1] On occasion we use alternate terminology such as "Objectives" in place of "Requirements", "Failure Modes" in place of "Risks", and "PACTs" – an acronym for Preventative measures, Analysis, process Controls, and Tests – in place of "Mitigations".

spacecraft experts, guided by their skill and past experience, generally make intuitive selections that are good. However, there is often some problematic area that use of the DDP model reveals. Visualization plays a prominent role by portraying the information in ways that make evident such problematic areas. DDP's capabilities in this regard are presented in the sections that follow.

3. Scrutinizing a candidate's status

3.1 Straightforward bar-chart visualizations

DDP uses straightforward bar chart visualizations to convey the status of an individual set of requirements and candidate decisions to achieve them. DDP generates bar charts to show:

- Requirements – each bar represents a Requirement, displaying its relative importance, and its degree of attainment in the current design.
- Risks – each bar represents a Risk, displaying its sum total expected impact on Requirements, both without any Mitigations (the worst case), and with the design's selection of Mitigations.
- Mitigations – each bar represents a Mitigation, displaying its expected risk-reducing effect.

An example of DDP's bar chart visualization of Risks' status is to be seen in Figure 1,. Each bar corresponds to a risk. The height of the bar indicates the sum total expected impact the risk causes – the height of the green portion indicates its sum total impact were no Mitigations to be selected, while the height of the red portion indicates its sum total impact taking into account the effects of the design's selection of Mitigations. The number beneath each bar is a reference to the corresponding risk in a list of such.

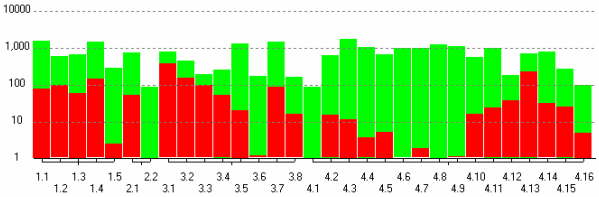

Figure 1. Bar chart of Risks' status

The bar chart shows that some of the Risks remain at relatively high levels, while some others have been reduced to relatively low levels. Note that the vertical scale is logarithmic, so the disparities are quite pronounced. This simple visualization is adept at revealing unbalanced treatment of risks: excessive resources expended to reduce some risks to tiny levels, while other significant risks remain relatively unaddressed.

DDP is also able to track *ranges* of quantitative values (e.g., the experts' estimates of effectiveness of requirements inspections at uncovering ambiguously worded requirements might range from 60% to 80%). A variant of the bar chart display showing such ranges is seen in Figure 2.

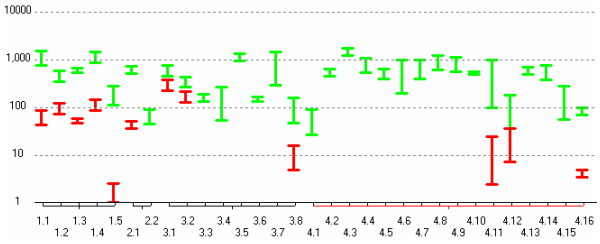

Figure 2. Ranges of risks

This kind of visualization is useful to point out the consequence of uncertainties in the input data on the computed values.

DDP also offers the capability to sort the bars (e.g., in descending order of mitigated risk), and to group and/or color-code the bars with respect to user-defined partitioning of risks into *categories*. Again, these are straightforward kinds of manipulations typical of bar chart displays.

3.2. Alternate visualizations

We have also experimented with using a form of TreeMap visualization [2] to present status information. An example appears in Figure 3.

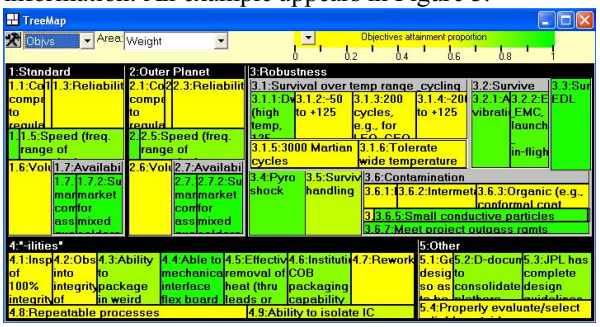

Figure 3. Treemap display of requirements

As is common in using TreeMap like displays, position, size and color are used to convey different aspects of the information. Here these are used as follows:

- *Position* represents the *requirements hierarchy*; sub-requirements of the same grouping appear as smaller rectangles within a larger rectangular area representing that grouping.
- *Size* represents the *relative importance* of the requirement or group of requirements it represents.
- *Color* represents the *attainment status* of that requirement, where color ranges from yellow, indicating a proportion attained of zero, to green, indicating a proportion attained of 1.0.

0-7695-2711-6/06 $20.00

Thus a large yellow area would indicate an important but unattained requirement. This makes it easy to see the extent to which a given design satisfies the important requirements. This same information can be gleaned from a bar chart, but the TreeMap display often appears more effective as a means to convey this information.

For understanding how risks are distributed across the likelihood / impact (a.k.a. severity) dimensions, we use a 2-D chart layout, as seen in Figure 4.

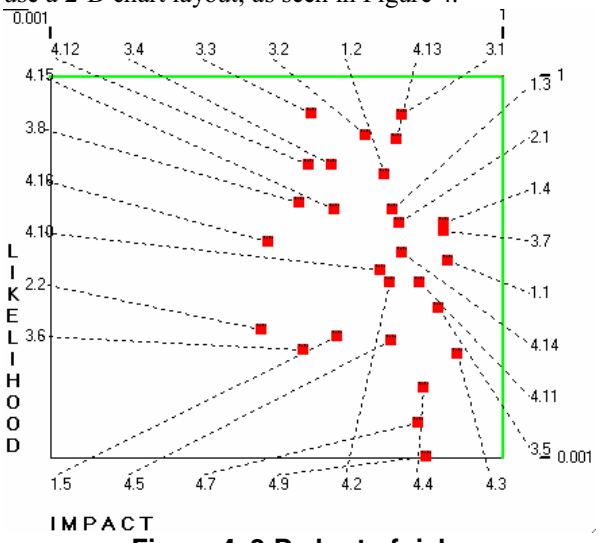

Figure 4. 2-D chart of risks

Each risk is represented by a small black square, located with respect to the chart's two axes, representing likelihood (vertical) and impact (horizontal). The dotted lines from each black square lead to the risk reference number, identical to that used in the bar chart display.

4. Comparing the details of alternates

When deciding among alternate design candidates it is useful to be able to scrutinize their details. For typically sized DDP models, computing and drawing the display takes less than a second on a 1GHz laptop for a design's selection of Mitigations. This makes is easy for users to try what if scenarios of alternates, and use the displays shown in the previous section to scrutinize each one.

In addition, DDP can generate bar charts that display the *differences* between a pair of designs. An example is seen in Figure 5.

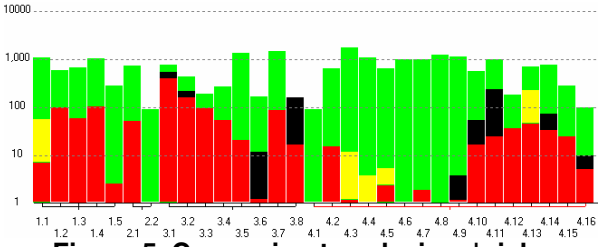

Figure 5. Comparing two designs' risks

This figure shows a comparison of the risk status of two designs. The baseline design is the one shown earlier in Figure 1; an alternate design is being compared against it. Black indicates where a risk has increased with respect to the baseline, and yellow where a risk has decreased.

For comparison among several designs, we have found that a Kiviat chart is appropriate to simultaneously display up to about half a dozen at once. Figure 6 shows an example.

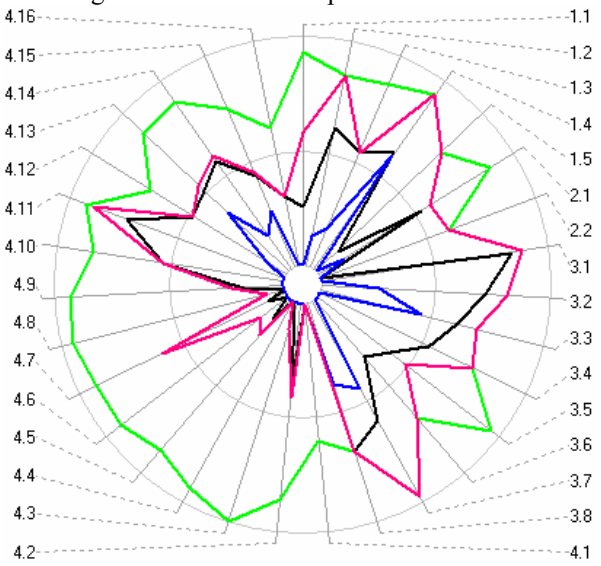

Figure 6. Kiviat chart of several designs' risks

On this chart each spoke represents a risk. Each design's status is indicted by a polygon whose edges are in a color corresponding to that design. The polygon's vertices are positioned along the spokes to indicate risk magnitude by distance from the center. As before, dotted lines connect risks to their reference numbers.

5. Understanding the solution space

Finding a desirable selection of Mitigations can be challenging, because of the number of Mitigations from which to select, and the convoluted nature of the way that Mitigations connect to Risks, and Risks connect to Requirements. If there are n mitigations, then there are in principle 2^n possible selections from

among them. To solve this, we implemented simulated annealing [16] within DDP, and use it to locate near-optimal solutions. We have also explored other forms of heuristic search: genetic algorithms and machine learning [6].

We use visualization to convey the results of such searches, as seen in Figure 7. This plots the result of an amalgam of searches, revealing the overall cost-benefit tradespace. Each of the approximately 300,000 individual points in the black cloud corresponds to a distinct selection of Mitigations. The DDP model has been used to calculate the cost and benefit of each such selection, and draw a small black point corresponding to the solution: cost determines horizontal position; benefit vertical position. The upper-left frontier of the cloud is thus the optimal boundary, also referred to as the Pareto front [19]. Note that while the simulated annealing search is designed to concentrate towards this optimal boundary, we plot a point for *every* selection investigated by the search, not just the near-optimal points on that boundary.

For this paper we have annotated the plot to indicate distinct regions on the Pareto front. Points within the interior are all inferior to more optimal solutions, of course. If the budget is low, the optimal solutions fall within the region where small amounts of additional funding can lead to radical improvements (i.e., better attainment of Requirements). Conversely, if the budget is high, optimal solutions fall within the region where a law of diminishing returns operates. The ideal is to be somewhere in the sweet spot region. If the budget is too small to allow this, such a plot can motivate either a request for a budget increase, or serious consideration of descoping (reducing expectations) to be more in line with the available budget.

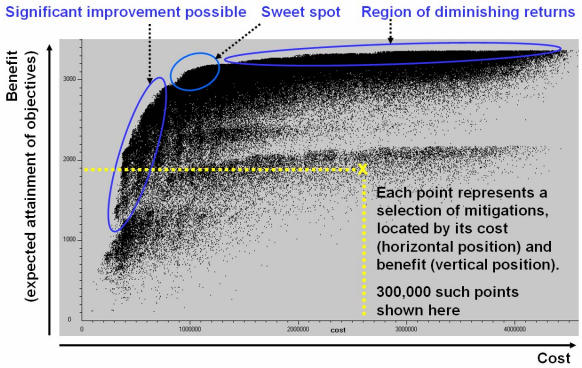

Figure 7. Cost-benefit tradespace chart

Sometimes the cost-benefit tradespace has a much more granular structure, representing different regimes of solutions at different expenditure levels.

Overall we find this kind of plot cogently reveals the overall cost-benefit tradespace, information that supports managerial decision making (e.g., can we afford this development? is the funding level appropriate to the problem?).

6. Exploring the solution space

Early in a project lifecycle a plethora of options are available. We find this to be the case when we scrutinize the results of our use of heuristic search to reveal the cost-benefit tradespace. Even when the users narrow their attention to a relatively small area in the tradespace there can be thousands of alternative solutions. This is illustrated with reference to the cost-benefit tradespace shown in Figure 7. Suppose we focus on a neighborhood of interest within the sweet spot characterized by solutions costing no more than $1,000,000, and by attaining at least 95% of the maximum benefit attainable within that region – see Figure 8.

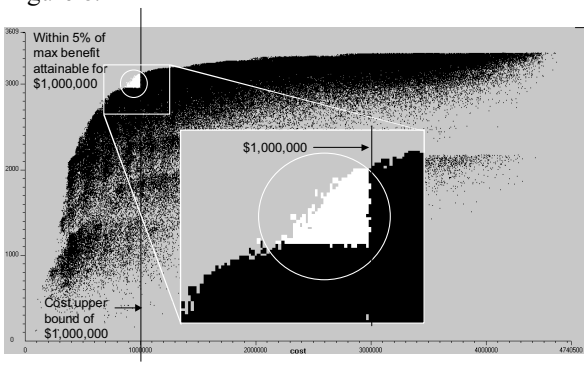

Figure 8. Neighborhood of interest

Within the dataset that gave rise to this picture, there are over 3,000 solutions, each of which is a distinct selections of Mitigations. Many of these will be similar – from one to the next, they may differ by only one or two low-cost, low-benefit Mitigations. However there may be some radically different solutions present within that same region. We have experimented with several techniques to explore such regions. Again, we use a mix of computational power to automate the exploration, and appropriate use of visualization to reveal interesting implications, discussed next.

6.1 Determining Key Decisions

A desirable thing to know is which decisions are *key* that is, make a significant difference to the outcome. In our context, an individual decision is represented as whether to select a Mitigation. Recall that each Mitigation represents a design or development option, so these decisions translate into design or development choices.

Methods for identifying key decisions have been the focus of one of this paper's authors (Menzies) for several years [17]. In collaboration we have studied their use within DDP [11]. Briefly, the study involved several iterations between the DDP tool, and the treatment learning tool. Each iteration revealed some additional key decisions – Mitigation selections and avoidances (ones to *not* select). In our study, the process terminated when decisions about one-third of the Mitigations had been identified, the result of which was convergence on a small area within the cost-benefit tradespace. The remaining two-thirds of the Mitigations turned out to be such small contributors to variation that the treatment learning approach could not discern any particularly key remaining decisions among them.

We use visualization to convey the overall consequences of this, as seen in Figure 9.

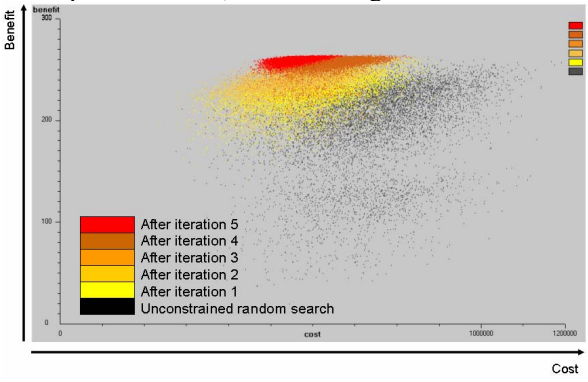

Figure 9. Convergence as iterations identify key decisions

The convergence towards the compact (red colored) zone of high benefit solutions is strikingly apparent. (Note that this figure is generated from a different model to that underpinning the other cost-benefit figures, so do not try to compare them.) The net result is knowledge of how to get to a desired area in the search space by having identified the subset of the key decisions and how to make them. The visualization serves to convince the viewers of the efficacy of that key decision set.

6.2 Understanding individual decisions' contributions

We have also experimented with a purely visualization-based approach to understanding the contribution of individual decisions [7].

Given the set of points of that constitute a cost-benefit tradespace, for a Mitigation of interest, we color each point one color (black, say) if that point corresponds to a solution not involving use of that Mitigation, and another color (yellow, say) if it does involve use of that Mitigation. An example is in Figure 10.

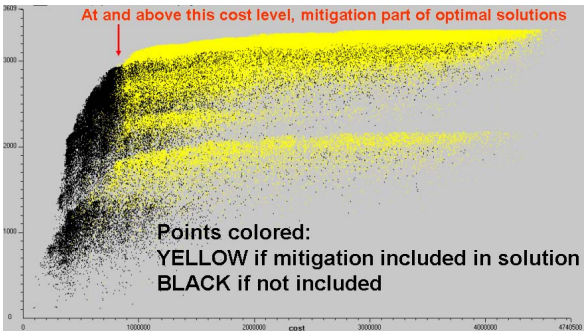

Figure 10. Contribution of an individual mitigation

The broad swathe of yellow points that dominate a large fraction of the Pareto front indicate that the chosen mitigation is key to nearly all optimal solutions above a certain cost level.

Repeating this for each Mitigation we can gather a snapshot of their contributions, seen in Figure 11. The four groupings therein correspond to the four major user-defined categories into which these 58 Mitigations were organized.

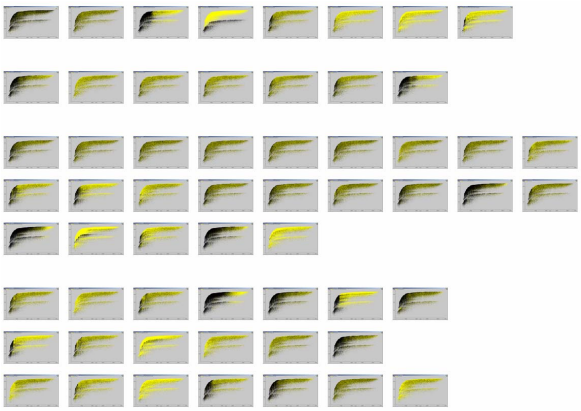

Figure 11. Snapshots of each Mitigations' contributions

We have since learned of the visualization tool ATSV [20] which uses a combination of techniques, of which color is just one, to show multiple dimensions of the attributes of a design simultaneously. We are currently investigating its use on our datasets, and the preliminary results so far are very promising.

6.3 Identifying interesting alternatives

We have also exploring means to distill information from the many solutions that lie within a neighborhood of interest. Our approach to this has been based on a definition of a metric of similarity between two solutions (i.e., selections of Mitigations). This metric is user-defined in terms *other* than overall cost and/or

benefit of a solution, since all the solutions are within a small neighborhood of similar costs and similar benefits. One useful metric might be based on the cost *profile* – how the costs fall into major categories. For example, in a hardware-centric study we did, there were categories of design, fabrication, assembly and test; two solutions that had the same overall cost but allocated that cost very differently between those categories would be very dissimilar.

Using such metrics we have explored the use of two techniques: (1) Search for maximally dispersed solutions within the neighborhood of interest. The idea of this is to yield a small number of interestingly distinct solutions [10]. (2) Search for clusters of similar solutions. The idea of this is to yield a small number of interestingly distinct clusters, within each of which all the solutions are relatively similar to one another [15]. For both of these, we again turn to visualization to present the results.

The visualization of a modest number of dispersed solutions is seen in Figure 12**.** The union of Mitigations involved in one or more of those solutions form the rows. The grid at the left is used to indicate the distinct solutions, one per column. A black (white) cell indicates that the Mitigation of that row is included (not included) in the solution of that column. Mitigations that are common to all solutions have been filtered out of this table to be listed separately (not shown here), so what remains is a portrayal of the differences among solutions. In this case the Mitigations are sorted in descending order of their cost. From this, it is easy to see that 8 out of 10 of the solutions include a $200,000 Mitigation, while 2 of them avoid its use. This is a significant difference given that the sum total cost of each solution is capped at $1 million.

The visualization of clusters is similar, seen in Figure 13. Rows correspond to Mitigations, while in this portrayal columns correspond to *clusters*. Since a cluster itself comprises multiple solutions, a given Mitigation may be involved in none, some or all of the solutions in a cluster. This is indicated by shading the square – white means not involved in any of the solutions within that cluster; black means involved in all the solutions within that cluster, and intermediate shades of grey denote intermediate levels of involvement.

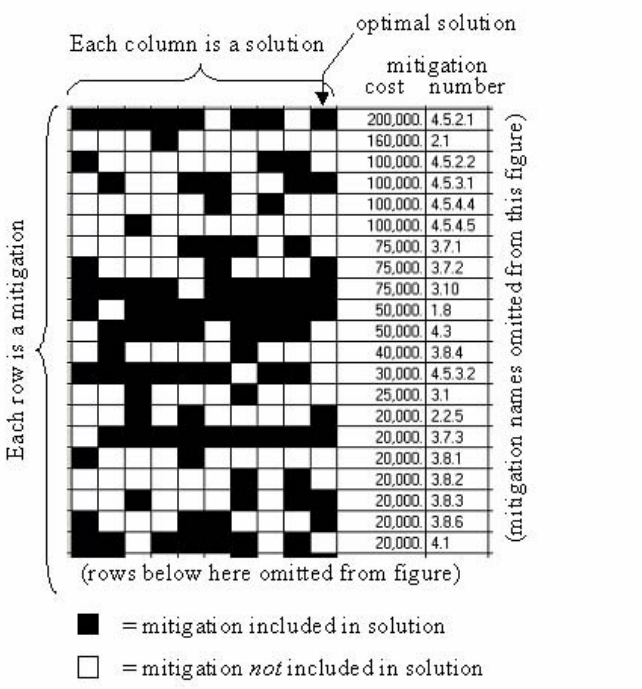

Figure 12. Visualization of 10 dispersed solutions

Figure 13. Visualization of clusters of solutions

7. Investigations of research and technology portfolios

In addition to applications of DDP to assist the infusion of *individual* technologies, in some cases there have been DDP applications that have focused on entire *portfolios* of technologies, the aim being to make a selection of a set of technologies to pursue. The first of these was conducted by JPLer David Tralli, who used DDP to assist activity selection across an entire program of NASA Earth Science Missions [21].

We have since used this same approach in a pilot study of the connections between the needs of software Independent Verification and Validation (IV&V) practitioners and a related software assurance research program containing multiple research efforts [12]. The purpose was to gauge how well the research program matched practitioner needs. In our pilot study, based on a partial set of data, we used DDP's visualization of the topology of connections between Requirements, Risks and Mitigations to present the data – the result is seen in Figure 14.

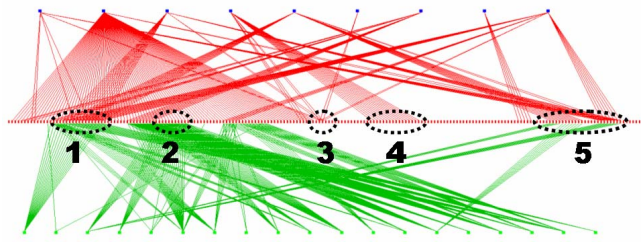

Figure 14. Topology of Needs-Areas-Researchers

The top row represents 9 IV&V practitioners. They were asked to express their needs in terms of the 198 leaf nodes in the software area of the ACM Computing Classification System [1]. These 198 leaf nodes form the middle row, and red lines connect practitioners with their expressed needs. The bottom row represents 19 researchers who were asked to express their research in terms of the same 198 leaf nodes in the software area. The green lines connect researchers to the areas they work in.

We have annotated the DDP-generated chart with 5 ellipses, each exemplifying a different phenomenon:

1. Very good overlap between areas of need shared by multiple practitioners and those same areas included in multiple researchers' activities.
2. An area of potentially over-addressed needs. Only one practitioner has expressed needs in these areas, yet multiple researchers have activities in these areas.
3. Unaddressed needs shared by several practitioners.
4. Unaddressed needs of a single practitioner.
5. Good overlap – areas shared by several practitioners and covered by several researchers.

Furthermore, our data included quantitative estimates, of the relative importance of a practitioner's needs, and the relative levels of effort of a research's activities.. This quantitative data allowed us to utilize DDP's risk calculations, but in this model instead of Requirements impacted by Risks we had we had Practitioners with needs for improvements in Areas of Computer Science, and instead of Risks effected by Mitigations, we had Researchers potentially contributing to progress in those Areas of Computer Science.

To show the quantitative aspects, we couple the topology visualization with the bar chart visualization. This is seen in Figure 15.

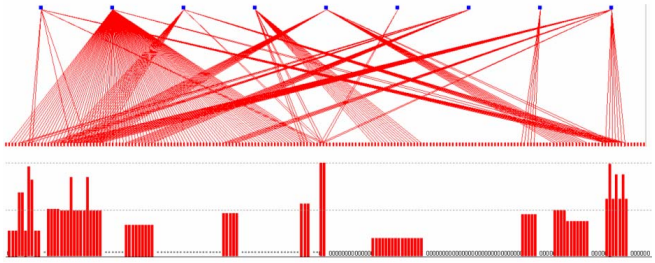

Figure 15. Topology coupled with bar chart

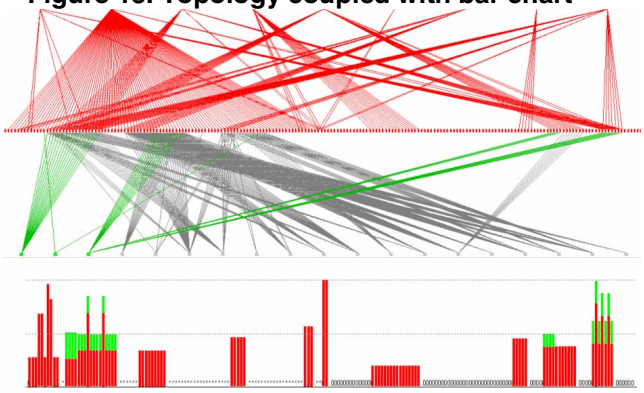

Figure 16. Fulfillment of needs from several research efforts

The red bars indicate sum total need in each of the areas, the higher the bar, the more need; again, bar heights are with respect to a log scale, so the differences are quite pronounced.

The extent to which researchers' activities contribute to the practitioners needs can be calculated and displayed, as seen in Figure 16. This figure shows the status when the left three research activities are selected, identifiable by their connecting lines colored green. DDP has calculated their combined effect at meeting practitioners' needs. The bar chart colors of red and green indicate, respectively, unfulfilled need, and need fulfilled by the currently selected researchers.

This topology visualization appears very well-suited to revealing these phenomena in this particular model, particularly when coupled with the bar chart display.

8. Conclusions and related work

The purpose of this paper has been to illustrate the application of software visualization to the earliest phases of the development lifecycle.

We have illustrated this with examples taken from our studies of spacecraft technologies and systems, spanning software, hardware and combinations of both. All figures in this paper are generated by this software, and all are based on actual models constructed in the course of our work.

8.1 Observations

- The requirements phase, when many key decisions are made on the basis of partial information, is a ripe area for use of visualization.
- We have not found one single visualization technique that will serve all purposes – instead we use a mix of several.
- We make use of relatively simple, commonplace visualizations: bar charts, 2-dimensional scatter plots, TreeMap, connection graphs and tabular formats. These have proven sufficient for many of our needs.
- Which visualizations to use and when is closely coupled to the model and data. Our highly quantitative model favors use of bar charts etc. to present status of items, and a fixed layout for the topology of the connections in our model (e.g., Figure 14) suffices. By way of contrast, the goal graphs of van Lamsweerde and Letier [22] are more general graph structures, presenting greater need for graph layout capabilities.
- Visualization and computation can complement one another (e.g., search for interesting alternatives was followed by visualization to show their makeup – section 6.3)

8.2 Related work

Karlson & Ryan's pioneering work on requirements prioritization [14] looked into the challenges of selecting the set of requirements to go into the next iteration of development or release of a product. The approach is based on gathering estimates for each requirement of its cost and benefit. Visualization in the form of a 2-dimensional chart presents the cost vs. benefit position of individual requirements, thus allowing users to select accordingly. Similar approaches are seen in the Win Win project [3] which supports multiple stakeholders to identify conflicts between their respective evaluations of requirements, and to locate feasible solutions that are mutually satisfactory combinations of requirements. Again, visualizations in the form of 2-dimensional charts are used, provided by the automated aids built to support this approach [13]. Regnell et al use of bar and pie charts for similar purposes [18]. By way of contrast, we do not feel able to ascribe directly to a requirement its cost and benefit – rather, we use our three-layer Requirements, Risks and Mitigations model to capture the more intertwined dependencies that we find arise in many of our studies. Also we deal with larger numbers of items – many dozens, sometimes hundreds – for which there is need for additional mechanisms to sort, elide, filter etc. Again, relatively simple such mechanisms appear to suffice.

Our display of the Pareto front to visualize the cost-benefit tradespace is commonplace in the typically *non*-software design optimization world [19]. The use of heuristic search techniques as a tool of optimization of software engineering problems is discussed in Clarke et al [4]. That article surveys past applications of heuristic search in areas of test data generation, module clustering and cost/effort prediction, and considers potential applications in additional areas. Interestingly, one of the areas they consider is the aforementioned requirements prioritization problem. Their focus, however, is on matching software engineering problems to heuristic search methods in order to be able to apply those methods. They do not continue to the point where the search results must be presented to users, and so are not motivated to consider issues of visualization in support of this.

9. Acknowledgments

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 NASA's Exploration Systems Mission Directorate. 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.

Research and development into the DDP risk-based model has been supported by NASA's Office of Safety and Mission Assurance (FDPP Program and IV&V ARRT task), Code R (ECS program), and ESMD Integrated Modeling and Simulation.

.

10. References

[1] ACM Computing Classification System [1998 Version] http://www.acm.org/class/1998/

[2] Bederson, B.B., Shneiderman, B. and Wattenberg, M. "Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies", *ACM Transactions on Graphics (TOG)*, 21 (4), October 2002, pp. 833-854.

[3] Boehm, B., Bose, P., Horowitz, E. and Lee, M. "Software Requirements as Negotiated Win Conditions", *Proceedings 1st International Conference on Requirements Engineering*, Colorado Springs, Colorado, IEEE Computer Society 1994, pp. 74-83.

[4] Clarke, J., Dolado, J.J., Harman, M., Hierons, R., Jones, B., Lumkin, M., Mitchell, B., Mancoridis, S., Rees, K., Roper, M. and Shepperd, M. "Reformulating software engineering as a search problem", *IEE Proceedings - Software Engineering* June 2003, 150(03), pp. 161- 175.

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

[6] Cornford, S.L., Feather, M.S., Dunphy, J.R., Salcedo, J. and Menzies, T. "Optimizing Spacecraft Design – Optimization Engine Development: Progress and Plans", *Proceedings of the 2003 IEEE Aerospace Conference*, Big Sky, Montana, March 2003, pp. 7-3361 – 7-3368.

[7] Feather, M.S. "Towards Cost-Effective Reliability through Visualization of the Reliability Option Space", *Proceedings of the 2004 Annual Reliability and Maintainability Symposium (RAMS)*, Los Angeles CA, January 2004, pp. 546-552.

[8] Feather, M.S. and Cornford, S.L. "Quantitative Risk-Based Requirements Reasoning" *Requirements Engineering* (2003) 8: pp. 248-263.

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

[10] Feather, M.S., Kiper, J. and Kalafat, S. "Combining Heuristic Search, Visualization and Data Mining for Exploration of System Design Spaces", *14th Annual International Symposium Proceedings of INCOSE,* Toulouse, France, June 20-24 2004.

[11] Feather, M.S. and Menzies, T. "Converging on the Optimal Attainment of Requirements", *Proceedings of the IEEE Joint International Conference on Requirements Engineering,* Essen, Germany, September 9-13, 2002, IEEE Computer Society, pp. 263-270.

[12] Feather, M.S., Menzies, T. and Connely, J.R. "Matching Software Practitioner Needs to Researcher Activities", *Proceedings of the 10th Asia Pacific Software Engineering Conference (APSEC 2003)*, Chiang Mai, Thailand, December 10-12, 2003. IEEE Computer Society, pp. 6-16.

[13] In, H., Roy, S. "Visualization issues for software requirements negotiation" *Computer Software and Applications Conference*, 8-12 Oct 2001, pp. 10-15.

[14] Karlsson, J. and Ryan, K. "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software*, Sept./Oct. 1997, pp. 67-74.

[15] Kiper, J.D. and Feather, M.S. "Mining Complex Requirements Specifications to Mitigate Risk via Clustering", *Proceedings, Workshop on Intelligent Technologies for Software Engineering (WITSE'04)*, Linz, Austria, September 20-25 2004, Austrian Computer Society.

[16] Kirkpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P. "Optimization by simulated annealing". *Science*, (13 May 1983), Volume 220, Number 4598, pp. 671–680.

[17] Menzies, T. and Hu, Y. "Data Mining for Very Busy People", *IEEE Computer* 36(11), November 2003, pp. 22-29.

[18] Regnell, B., Host, M., Nach och Dag, J., Beremark, P. & Hjelm, T. "An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software", *Requirements Engineering* 2001 6: pp. 51-62.

[19] Sen, P. and Yang, J-B. *Multiple Criteria Decision Support in Engineering Design,* Springer-Verlag, 1998.

[20] Stump, G. M., Yukish, M., Simpson, T. W., and Bennett, L. "Multidimensional Visualization and Its Application to a Design by Shopping Paradigm", *Proceedings 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, 2002, AIAA, AIAA-2002-5622.

[21] Tralli, D.M. "Programmatic Risk Balancing", *Proceedings of the 2003 IEEE Aerospace Conference,* Big Sky MT, March 2003.

[22] van Lamsweerde, A. & Letier, E.. "Handling Obstacles in goal-oriented requirements engineering". *IEEE Transactions on Software Engineering*, 26(10), 2000, pp. 978-1005.