Relating Practitioner Needs to Research Activities

Martin S. Feather
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr
Pasadena CA 91109-8099
Martin.S.Feather@Jpl.Nasa.Gov

Tim Menzies
Lane Dept. of Computer Science
& Electrical Engineering,
West Virginia University
PO Box 6109
Morgantown, WV 26506-6109
tim@menzies.us

Judith R. Connelly
NASA IV&V Facility
100 University Drive
Fairmont, WV 26554
Judith.R.Connelly@nasa.gov

Abstract

Many organizations look to research to yield new and improved products and practices. Matching practitioner needs to research activities is important to guiding research and utilizing its results.

We present an approach to matching needs (practitioner requirements) to solutions (research activities). A taxonomical classification scheme acts as intermediary between needs and activities. Expert practitioners express their needs in terms of this taxonomy. Researchers express their activities in these same terms. A decision support tool is used to assist in the combination and study of their expressions of needs and activities.

The approach is demonstrated in the area of software assurance. The ACM Computing Classification System[1] (1998) is used as central taxonomy. In an experiment, 9 individuals' expressions of practitioner needs, and 19 individuals' expressions of research activities are combined. The value of utilizing a decision support tool to gain insight into overlaps (and gaps) within this combined data is exhibited.

Paper type: Research

[1] The ACM Computing Classification System [1998 Version] is Copyright 2002, by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permission to republish from: Publications Dept., ACM, Inc. Fax +1 (212) 869-0481 or E-mail permissions@acm.org.

Keywords: Technology transfer, Technology Infusion, Requirements Prioritizing, Requirements Elicitation, Tool Support for Requirements Engineering, Software Assurance

1. Introduction

Many benefits derive from successful technology transfer (the flow of ideas from research to widespread practice) and technology infusion (the adoption and use of research results by specific organizations). However the low rate at which these generally occur is of continual concern. Almost a decade ago, Zelkowitz's 1996 paper [Zelkowitz 1996] reported on a study of software engineering technology infusion within NASA. Many of his observations and insights remain valid today. He in turn references work from a decade earlier – Rogers's "Diffusion of Innovation" book of 1983 [Rogers, 1983].

These same concerns have recently risen to prominence within the Requirements Engineering community – [Kaindl et al, 2002] "*... summarises, clarifies and extends the results of two panel discussions, one at the Twelfth Conference on Advanced Information Systems Engineering (CaiSE'00) and the other at the Fourth IEEE Conference on Requirements Engineering (ICRE'00)*". As observed by Davis and Hickey in their "Viewpoint" article [Davis & Hickey, 2002], the subject should be of special interest to requirements engineering. Not only is the RE field subject to the same concerns, but also the understanding of customer needs in order to better target new products, including research itself, is at the very heart of what Requirements Engineering is supposed to do.

In this paper our focus is on the matching of customers' (software engineering practitioners) needs to researchers' activities. We examine this issue at the broad level of an entire program of research. We present an approach that allows scrutiny of an entire research program (*multiple*

individual research efforts) and *multiple* practitioners' needs. The intent is to yield insights for organizations and agencies in making their decisions of how to prioritize areas of research for support. It is also intended to yield insights for researchers, by revealing relatively unexplored areas of need.

The paper is structured as follows:

Section 2 considers the challenges of elicitation and representation of the information (practitioner needs and researcher activities). It presents the key idea of using a taxonomical classification scheme as the intermediary between practitioner needs and research activities. It introduces our demonstration of this approach in the area of software assurance.

Section 3 presents the second key step, the use of a decision support tool with which to explore the non-trivial amount of information that our demonstration involves.

2. Information elicitation and representation

Our objective is to be able to compare and reason about practitioners' needs for research advances, and researchers' activities that are intended to lead to such advances. This section considers the challenges of:

- information elicitation - how do we ask practitioners to express their needs, and researchers to describe their activities?, and
- information representation – how do we represent the answers we gather?

Because our focus is at the level of an entire research program, we need a vocabulary that is common to both practitioners and researchers. We also need the ability to accommodate expressions of multiple needs and activities, at varying levels of detail and emphasis.

The key to our solution is the use of a taxonomical classification scheme as intermediary between practitioner needs and researcher activities. We make several assumptions about this scheme, namely that it

- already exists,
- is understood by both practitioners and researchers,
- spans the range of concerns involved, and
- goes down to a sufficient level of detail to distinguish among different practitioner needs and different research activities.

We use a real-world experiment for illustration as we elaborate upon these issues. The subsections that follow introduce our experiment, and then return to the issues of elicitation and representation.

2.1. Experimental area – software assurance

We have a special interest in the area of *software assurance*, the planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures. (Note that activities in this definition refer to practitioner activities, not research activities.)

Examples of such activities are: code inspections, unit tests, design reviews, performance analyses, construction of tracability matrices, etc. Research that may yield advances of relevance to software assurance include refined inspection techniques, empirical studies that indicate the relative costs and benefits of various methods, advances in analysis techniques such as model checking, adaptation of methods such as probabilistic risk assessment to software problems, measurement and metrics studies, etc. NASA has an ongoing program that encompasses these kinds of research, reported annually in the form of the NASA Office of Safety and Mission Assurance's Software Assurance Symposium [NASA-OSMA-SAS 2001] [NASA-OSMA-SAS 2002].

We have conducted a substantial experiment that investigates matching expressions of software assurance practitioners' needs, and software assurance researchers' activities. We elicited expressions of practitioner interest from 9 individuals, and expressions of research activities from 19 individuals.

2.2. Taxonomical Classification Scheme

We selected the ACM Computing Classification System (1998) [ACM 1998] as our taxonomical classification scheme. This meets our assumptions in that it already exists and is understandable by both practitioners and researchers. It is crafted to cover the whole area of computing literature, so indeed spans the range of concerns that arise in software assurance. It goes down to a fairly detailed which we hypothesized would suffice for our purposes level, e.g., one of the leaf nodes is "Model checking" (within category D.2.4 Software/Program Verification).

2.3. Information elicitation

We planned to ask practitioners to express needs for advances that would help software assurance in terms of the ACM taxonomy. Likewise, we planned to ask researchers to express research activities in terms of this same taxonomy. However, the taxonomy is quite large – there are well over 1,000 leaf nodes. If we were to insist that each expression of interest/activity be stated in terms of leaf nodes, this would make broad ranging needs/activities, which encompass many such leaf nodes, very cumbersome to express.

Our solution to this dilemma was to ask for expressions of interest/activity stated in terms of nodes at any level of choosing – leaf node or not. Thus if a practitioner saw the need for advances in the level 3 category D.2.8 Metrics, but did not see the need to distinguish between the elements in that category (Complexity measures, Performance measures, Process metrics and Product

metrics), then we would allow that practitioner to express interest with respect to that non-leaf-node category. Similarly, if a researcher felt that a research activity contributed to the whole of the level 2 category D.4 OPERATING SYSTEMS, then this could be expressed with respect to that node in the taxonomy tree.

We came to realize that an individual might have interests/activities that relate to several nodes, but not of equal weight. For example, a practitioner might see the need for advances to be made in both D.2.8 Metrics, and D.4 OPERATING SYSTEMS, but put (say) twice as much weight on the former as the latter. Similarly, a researcher may estimate that a research activity contributes to several areas to differing extents, and so would correspondingly weight expressions of contribution.

A typical[2] expression is the following:

D.2.1 Requirements/Specifications: *Methodologies* **(2)**
D.2.5 Testing and Debugging: *Debugging Aids* **(1)**
D.2.5 Testing and Debugging: *Monitors* **(1)**
D.2.8 Metrics **(3)**

Each line references a node somewhere in the ACM Computing Classification Scheme, and concludes with a relative weighting in parentheses. Thus twice as much weight is being given to *Methodologies* within D.1.2 than is given to *Debugging Aids* within D.2.5, or *Monitors*, also within D.2.5. Note that D.2.8 Metrics is an entire category, which has been weighted as triple that of *Debugging Aids* and of *Monitors*.

2.4. Information Representation

The information representation that follows naturally from our elicitation method is to attach the weighted expressions of practitioner needs, and of researcher activities, to nodes within the tree representation of the ACM Computing Classification System.

Imaging a picture of the entire tree, annotated at various nodes with these expressions of needs and activities. It is easy to imagine that there could be considerable variations among the levels in the tree at which these various expressions are attached, and indeed the data we gathered exhibits this diversity.

2.5. Information Combination

Having gathered the information and attached it to the tree representation of the ACM CCS, the next step is to decide how to combine the information. The objective is to gain insight from the combined data into the overall status of research needs vs. activities.

The elicitation and representation steps result in information attached at various levels throughout the tree. For example, suppose a practitioner had indicated interest

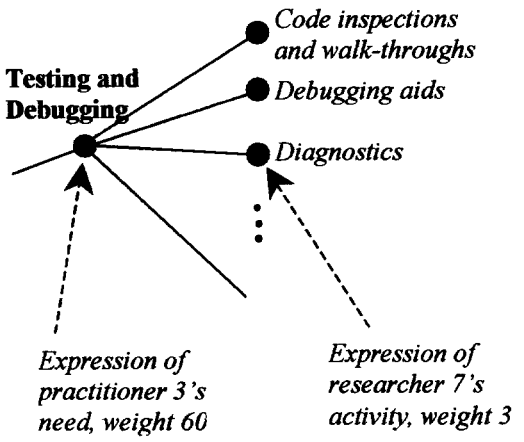

Figure 1 – mismatched levels of expression

in the entire category D.2.5 of **Testing and Debugging**, while a researcher had indicated activity in just the one specific area of *Diagnostics* within that category (Figure 1). Intuitively, the researcher activity is making only a partial contribution towards the practitioner need. This dispersal of information makes comparisons difficult.

Our solution is to percolate all the weighted expressions of needs and activities down to the leaf nodes of the tree. For example, a practitioner's expression of need attached to the **Testing and Debugging** node percolates down to expressions of need by that same practitioner for each of the leaf nodes of *Code inspections and walk-throughs, Debugging aids, Diagnostics*, etc. In doing so, at each level an expression's weight is subdivided equally among the children of that level. The net result will be that the resulting expression weights assigned to the leaf nodes sum to the expression weight originally assigned to the non-leaf node.

There is a choice of how to subdivide those weights – an alternative to the above scheme would be to assign an equal amount could be given to every leaf node. These lead to different weight assignments to leaf nodes when more than two levels are involved and the tree is not perfectly balanced. For example, consider the tree (1 (1.1 (1.1.1, 1.1.2)) 1.2); a weight of 60 on node 1 percolated using the first scheme would yield

1.1.1's weight = 15
1.1.2's weight = 15
1.2's weight = 30

while the alternative scheme would yield

1.1.1's weight = 20
1.1.2's weight = 20
1.2's weight = 20

Our choice of the recursive subdivide-among-children scheme is based on the way in which we posed our question during information elicitation.

The net result of this is a tree whose leaf nodes are

[2] For disclosure reasons, the data reported in this paper has been deliberately skewed from the actual information that was gathered.

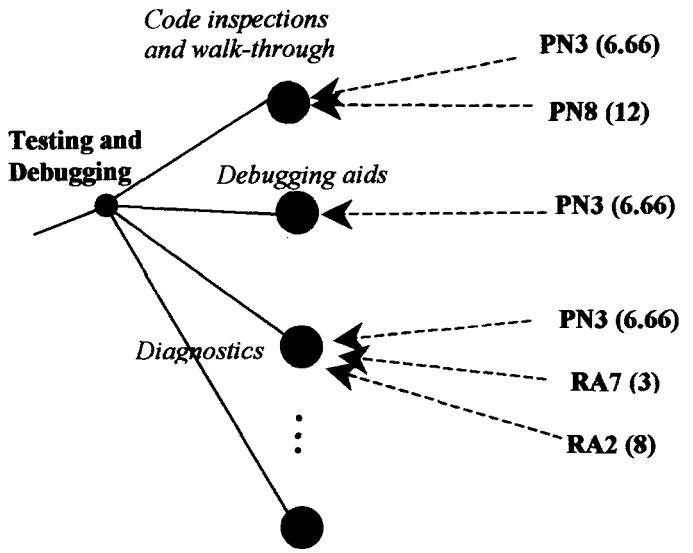

Figure 2 – Practitioners' weighted needs and researchers' weighted activities attached to leaf nodes

labeled by weighted expressions of needs (derived from the expressions of practitioners), and activities (derived from the expressions of researchers). A fragment the resulting tree for hypothetical data is pictured in Fig. 2.

As suggested in this picture, there can be a mixture of multiple practitioners' needs and/or multiple researchers' activities all at a single node. Furthermore, recall that the ACM Computing Classification System has well over 1000 nodes. Finally, recall that we have expressions of need/activities from 9 practitioners and 19 researchers. The net combination is a large tree, with multiple labels at many of the leaf nodes. Manual scrutiny of this large tree is infeasible. Instead, we make use of a decision support tool to aid us in gaining insights from this combined information, as discussed in the next section.

3. Use of a Decision Support Tool

This section presents the second key step, the use of a decision support tool with which to explore the non-trivial amount of information that our experiment involves. The tool we use is Defect Detection and Prevention (DDP), intended for risk-based decision making. DDP's primary purpose is to help experts assess the risk status of a system/subsystem/technology and plan a cost-effective set of risk mitigations. At first sight its selection for this task may seem counterintuitive. There is however an interesting analogy between DDP's approach to risk management and the task at hand. This yields some suggestions on how to interpret the information we have gathered, and also makes possible the use of DDP's mechanisms for visualization and reasoning over non-trivial amounts of information. The inspiration for this analogy came from JPLer David Tralli's inventive use of DDP to assist activity selection across an entire program of NASA Earth Science Missions [Tralli, 2003].

We first give some background in DDP, and then introduce the analogy that lets us apply DDP to the task at hand. The details of this application follow, considering the way in which the information was imported into DDP, the DDP-motivated and assisted interpretation of the practitioners' "needs" data, and finally the DDP-motivated and assisted interpretation of how the researchers' "activities" data contributes to meeting those needs.

3.1. Background to the Defect Detection and Prevention (DDP) tool

JPL deploys spacecraft in new and challenging situations, employing new technologies and designs to better attain mission objectives. Quality assurance must therefore adapt and extend best practices and lessons learned from past missions to each new spacecraft. Dr. Steve Cornford at JPL conceived of a quantitative model specifically to facilitate this [Cornford 1998]. His model, called "Defect Detection and Prevention" (DDP), was designed for application early in the lifecycle, when information is sparse, yet the capability to influence the course of the development to follow is large. Initial experiments that manually explored the utility of the DDP process had positive results. These led to development of custom software [Feather et al, 2000]. Supported by this software, DDP has been applied to assess the viability of, and planning for, the development of novel technologies and systems [Cornford et al, 2001], [Cornford et al, 2002].

The core idea of DDP is to relate three sets of information: *requirements* (a.k.a. *objectives*) – whatever the system is to achieve, *risks* (a.k.a. *failure modes*) – things that might possibly occur that can get in the way of attaining those requirements, and *mitigations* – options available for reducing the likelihood and/or severity of those risks. DDP uses a *quantitative* treatment of the relationships between information (*how much* a risk, should it occur, detracts from a requirement's attainment, and *how much* a mitigation, should it be applied, reduces a risk). Mitigations have costs (schedule, budget and other resources of concern to spacecraft, e.g., mass, power, volume). In almost all cases, the sum total cost of mitigations exceeds the resources available, so one of the primary purposes of DDP is to help arrive at a cost-effective selection of mitigations.

3.2. Analogy between risk-centric DDP and relating practitioner needs & researcher activities

Standard DDP applications deal with requirements and mitigations, using risks as intermediaries. Selection of mitigations leads to attainment of requirements (by quelling risks). For our present task of relating practitioners' needs and researchers' activities, there is a useful analogy in which requirements correspond to practitioners, mitigations correspond to researchers, and risks correspond to areas of computer science (see Fig. 3). More precisely, risks correspond to (lack of) advances in areas of computer science.

In standard risk-centric DDP, risks serve as a useful intermediary. A risk may impact multiple requirements to different extents; a requirement may be impacted by multiple risks to different extents; a mitigation may reduce multiple risks to different extents; a risk may be reduced by multiple mitigations to different extents. The same holds for practitioners, needs and researchers. A practitioner may have expressed the need for advances in multiple areas of computer science to different extents; advances in an area of computer science may have be needed by multiple practitioners to different extents; a researcher may have expressed that his/her activity contributes advances to multiple areas of computer science to different extents; advances in an area of computer science may be provided by multiple researchers' activities to different extents.

3.3. Importing the practitioner & researcher data into DDP

To import our data into DDP, each Practitioner became a separate DDP requirement, each Researcher became a separate DDP mitigation, and the ACM Computing Classification System tree became a DDP risk tree (the DDP tool supports tree-structured data).

Each practitioner's expression of need was percolated down to the leaf level, as outlined earlier in section 2.5. Each of the resulting weighted expressions of need was used to link the practitioner to the computer science area, scoring that link with the weight computed in the percolation process. Similarly, each researcher's expression of activity was percolated down to the leaf level, as outlined earlier in section 2.5. Each of the resulting weighted expressions of activity was used to link the researcher to the computer science area, scoring that link with the weight computed in the percolation process.

The needs and activities data was available as text files in a well structured format. It was a simple matter to write additional DDP code to process these files and create the appropriate DDP objects and links.

3.4. Interpreting the needs data

The risk analogy also gives a suggestion as to how to interpret the combined needs and activities data. We begin by focusing on the needs data.

3.4.1 Calculating the extent of a practitioner's needs

In risk-centric DDP applications, the extent to which a requirement is *unmet* is determined by the sum of risks impacting that requirement. By analogy, the extent to which a practitioner's needs are unmet is determined by the sum of the unachieved advances impeding the computer science area needs of that practitioner.

More precisely, in risk-centric DDP applications:

- a requirement may be impacted by risks;
- each such impact is scored by the proportion of the requirement that would be lost if that risk were to occur. For requirement Q and risk R we will write this as: Impact(R,Q)
- each risk's likelihood of occurrence is calculated from its a-priori likelihood and the risk-reducing effects of selected mitigations. For risk R we will write this as Likelihood(R)

Hence a requirement Q's "at-risk" measure is Σ (R $\in$ Risks): Impact(R,Q) * Likelihood(R)

So for our needs-activities task:

- a practitioner may be impeded by need for progress in computer science areas;
- each such impedance is scored by the proportion of the practitioner's weights given

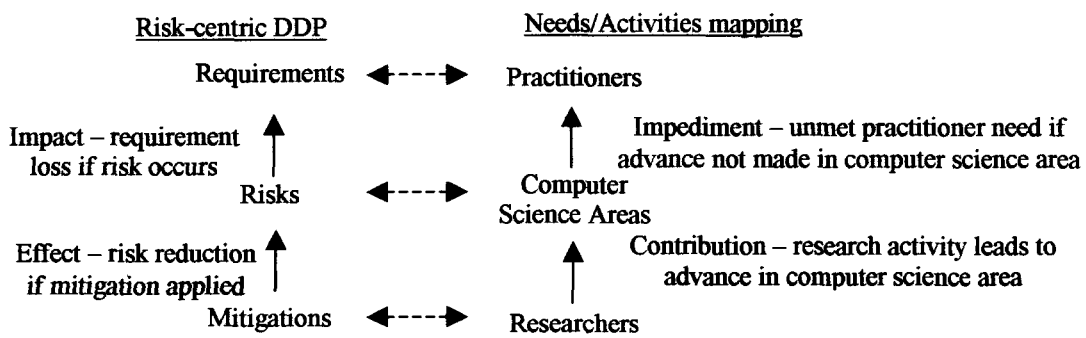

Figure 3 – Analogy between standard risk-centric application of DDP and mapping practitioners' needs to researchers' activities

to that area. For example, if a practitioner had identified three areas of need, and weighted them 15, 15 and 30, then the first and second of these would each score 0.25 (15 / (15 + 15 + 30)), and the third would score 0.5 (30 / (15 + 15 + 30)). For practitioner P and area A we will write this as: Need(P,A)

- each area has some potential for progress; the expected contributions of research activities in that area contribute to such progress. For an area A we will write this as Progress(A)

Hence a practitioner P's "unmet-need" measure is

Σ (A $\in$ Areas): Need(P,A) * (1 - Progress(A))

3.4.2 *Calculating the extent of an Area's needs*

As well as calculating the "at-risk" measure for requirements, risk-centric DDP also calculates the "total impact" measure for risks. For risk R, the formula is:

Σ (Q $\in$ Requirements) Impact(R,Q) * Likelihood(R)

In practice, some requirements are more important than others. This is captured by giving each requirement a "Weight", reflecting its relative importance. Taking this into account, the above formula becomes:

Σ (Q $\in$ Requirements):
Weight(Q) * Impact(R,Q) * Likelihood(R)

Similarly, in our needs-activities world, some practitioners are more important than others (for example, they may be responsible for the V&V of a larger program area). If we use "Weight" to capture relative importance in the same manner, the "total unmet need" measure for area A is:

Σ (P $\in$ Practitioners):
Weight(P) * Need(P,A) * (1 - Progress(A))

If there were no research activities taking place, then Progress(A) would equal zero for each area A, and the formula would simplify to

Σ (P $\in$ Practitioners): Weight(P) * Need(P,A)

Intuitively, this gives a quantitative measure of the practitioner need for advances in each of the computer science areas.

3.4.3 *Visualizing the "need" information using DDP*

The DDP tool automatically calculates the measures listed above, and provides several graphical means to visualize the information.

As an illustration, Fig. 4 is a screenshot taken from the DDP tool showing the leaf nodes in category D of the ACM Computing Classification System. In this view, the node names are listed in the same order as they appear as leaves in the tree. The length of the red bars to the left of the names indicate **extent of practitioner need** for research advances in that named area. (Note: the chart is generated from our collected data, modified somewhat to preserve confidentiality, but still indicative of the magnitude and complexity of the real data). The small-scale red lines to the right hand side of the image are a complete thumb-nail view of the entire D category, with a rectangle drawn around the small portion towards the top corresponding to the portion whose names are in view.

One of the useful DDP options is to sort these into descending order, from which we can see the areas most needing research advances. Figure 5 shows such a view, where the thumbnail shows that we have scrolled to partway down the sorted list.

3.5. Interpreting the activities data

We now consider how to interpret the researchers'

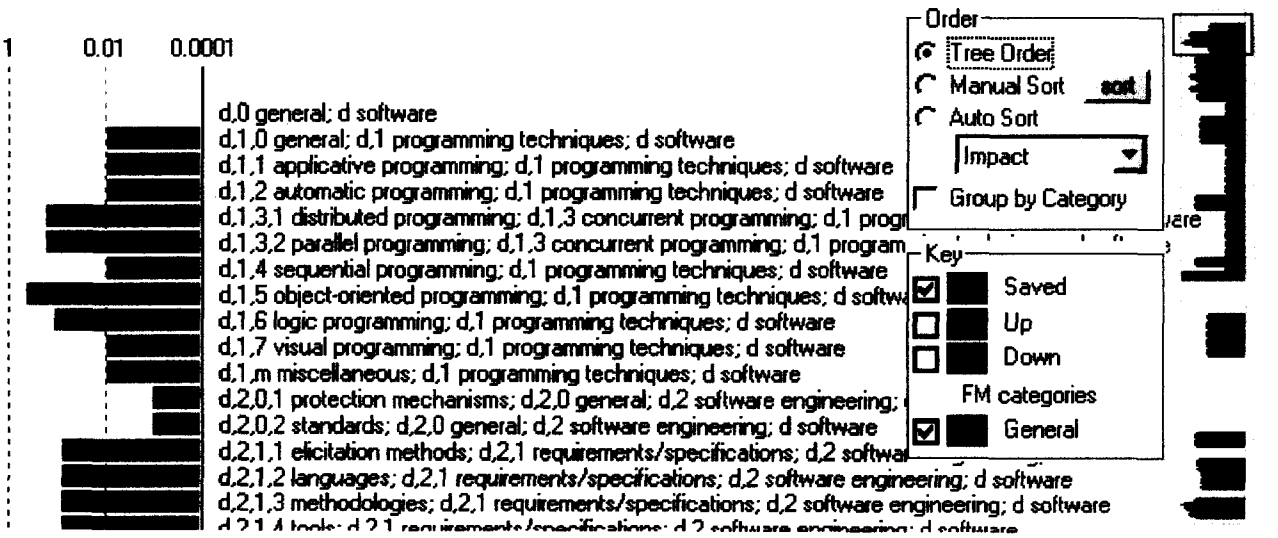

Figure 4 – DDP visualization of magnitude of practitioner needs in computer science areas

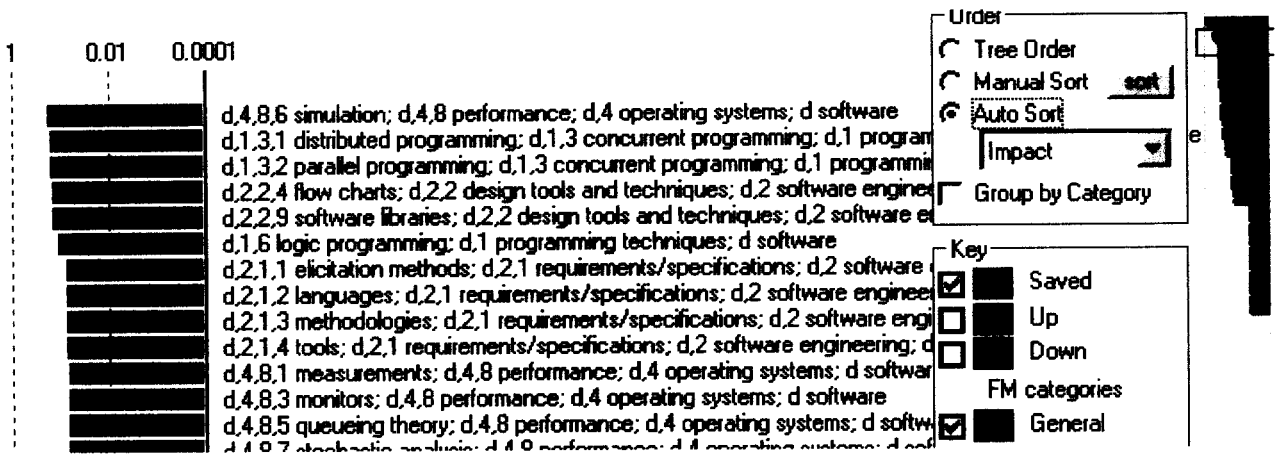

Figure 5 – DDP visualization of practitioner needs in computer science areas, sorted into descending order

activities data.

3.5.1 *Calculating the contribution of a researcher's activities*

In risk-centric DDP applications, mitigations reduce risks, and so lead to greater attainment of requirements. By analogy, researchers' activities contribute advances to areas of computer science, and so lead to meeting more of the needs of practitioners.

More precisely, in risk-centric DDP applications:

- a risk may be effected by mitigations. (Usually the effect is a reduction of either the risk likelihood or the risk severity; DDP also allows for the case that a mitigation makes certain risks worse. For the analogy used here, only risk reduction is relevant);
- each such effect is scored by the proportion by which the risk would be reduced if that mitigation were applied. For mitigation M and risk R we will write this as: Effect(M,R)
- each risk's likelihood of occurrence, Likelihood(R), is calculated from its a-priori likelihood and the effects of applied mitigations, thus:

A-Priori(R) * Π (M $\in$ Mitigations): (1 – Effect(M,R))

Intuitively, mitigations act like "filters", each filtering out some proportion of the incoming risks, with multiple filters arranged in series. For example, if one mitigation's effect is 0.9, it filters out 90% of the incoming risks, leaving 10% remaining. A second filter whose effect is 0.5 would filter out 50% of the risks that got through the first filter, leaving just 5% of the original risks remaining.

So for our needs-activities task:

- a researcher's activities may contribute research advances to areas of computer science;
- each such contribution is scored by the magnitude of the advances in that area. For researcher R and area A, write this as Contribution(R,A)
- in an area A the combination of multiple researchers' contributions within that area leads to a measure of Progress(A) =

1 – (Π (R $\in$ Researchers): (1 – Contribution(R,A)))

For example, if there were no researchers active in a given area, its Progress measure would be 0; if there was just one researcher active with Contribution=0.8, then its Progress measure would be (1 – (1–0.8)) = 0.8; if there were two researchers with Contributions 0.8 and 0.5, then its Progress measure would be (1 – (1–0.8)*(1–0.5)) = 0.9

This formula captures the intuitive notion of some overlap among the researchers activities within a given area. We discuss its validity in section 4.

3.5.2 *Visualizing the "contribution" information using DDP*

As was the case for the "needs" data, the DDP tool automatically calculates the measures listed above, and provides several graphical means to visualize the information.

As an illustration, Fig. 6 is a screenshot taken from the DDP tool showing the same leaf nodes of category D as appeared in Fig. 4, but now taking into account the contributions expected of the researchers' activities at meeting practitioners' needs. The green portions of the bars in this figure show **need that is expected to be met by the researchers' actitivies**. The length of the red bars in this figure indicate the extent of the **remaining unmet need for advances in each area**. Again, the thumbnail on the right side gives an overview of the entire D category.

From this kind of visualization it is possible to draw several kinds of insights at a glance, for example:

- *pure red bars* - the **lack of any research activity**

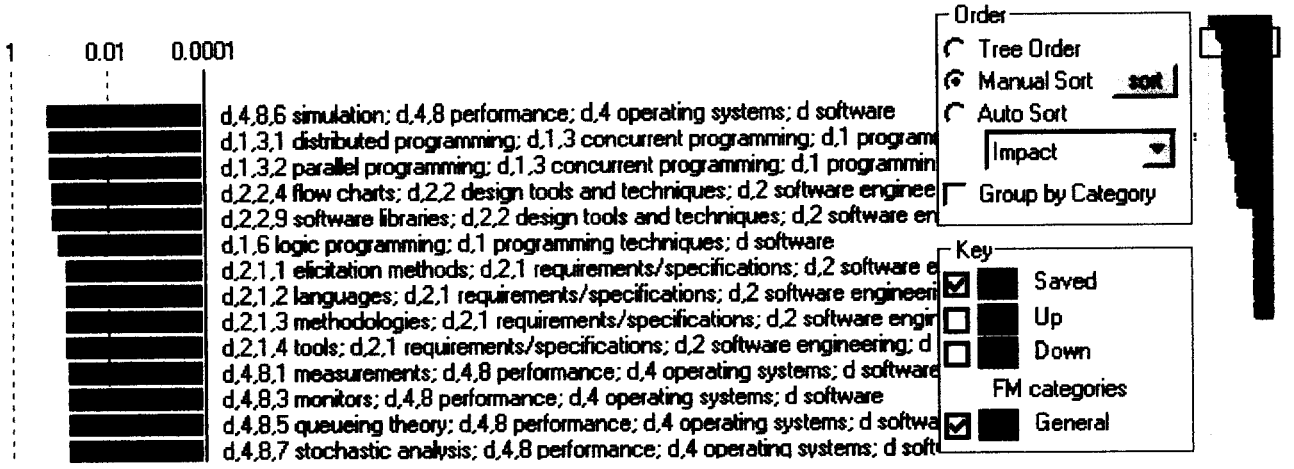

Figure 6 – DDP visualization of practitioner needs, and researcher activities contributing to meeting them

whatsoever in area d,1,3,1 (second from the top in the figure)

- *small to modest green bars* - **small to moderate amounts of research activity** in many of the visible areas
- *large green bars* - a **relatively large amount of research activity** contributing to the needs in area d.2.1.4
- *red and green regions in thumbnail* – overall, there is a good deal of **unevenness in fulfilling needs** extent to which areas of need are being met by e.g., there some areas of even greater unfulfilled need scrolled off the top, while there are some areas of relatively small need to which research is contributing significantly scrolled off the bottom.

It is of course possible to re-sort these based on residual need (length of the red bars) and so readily identify those areas of maximum unfulfilled need. This would be a useful indication of areas worthy of increased research.

3.5.3 Gaining insights from other DDP supported visualizations

Further insights can be gained from using DDP's other capabilities to summarize and visualize information. We give two examples.

Figure 7 shows a fragment of the screen showing a concise view of researcher activities. Each wide rectangle in the left column corresponds to an individual researcher. To the right are numerous smaller white rectangles labeled

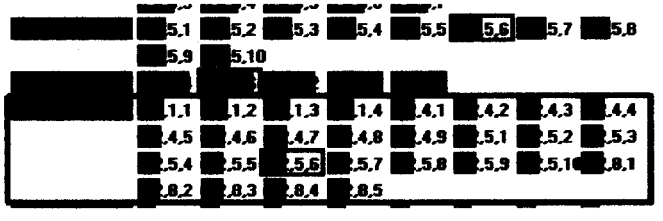

Figure 7 – A concise view of researcher activities

by nodes of the research area taxonomy. These indicate the areas in which that researcher's activities are expected to contribute. Turquoise smaller rectangles overlay those white rectangles, indicating the magnitude of the researcher's expected contribution to that area – the wider the turquoise rectangle, the greater the contribution. Finally, red borders highlight all the instances of the same area (the one over which the cursor – not visible – was positioned). Once the format of this display is understood, it is easy to gain insights at a glance. For example, the activities of researcher R5 are clearly concentrated in a small number of areas, while the activities of researcher number R6 are much more widely dispersed. For the area highlighted in red, we can see that all three of the researchers in the visible fragment are contributing to that area. This concise view is capable of showing all the researcher activities information on a single screen – hundreds of items in all. Analogous views exists in which:

- taxonomy areas are listed down the left, and researchers contributing to those areas are listed alongside,
- taxonomy areas are listed down the left and practitioners needing advances in those areas are listed alongside, and
- practitioners are listed down the left and taxonomy areas in which they have needs for advances are listed alongside.

DDP also provides the ability to study the tree information when the tree is only partially expanded (i.e., not necessarily all the way down to the leaf level). Fig. 8 shows such a view, where the D category has been expanded down to the first level for its sublevels 1,3,4 and m, and down to the second level for its sublevel 2 (d,2,0 ... d,2,m). The lengths of the bars are computed automatically by aggregating the information from the lower levels. This kind of view is well suited to seeing the "big picture", and allows the user to drill down into detail in the areas of the user's choosing.

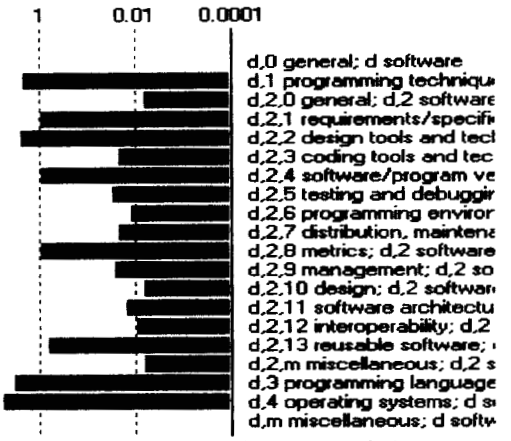

Figure 8 – View of partially expanded taxonomy

4. Discussion

4.1. Related Work

Our objective of matching practitioners' needs to researchers' activities is closely related to the classic requirements analysis problem of matching features of a to-be-developed product to customer needs. Representative work in this area includes:

Karlsson & Ryan's study of selection of requirements for software system developments [Karlsson & Ryan, 1997]. Their approach yields a 2-dimensional "cost-value" diagram in which each requirement is plotted as a point located according to its customer value in one dimension, and cost of implementation in the other dimension.

Kulik & Macdonald's approach to classifying project requirements into the major categories of "Add Value", "Must Do", "Nice to Have" and "Defer" [Kulik & Macdonald, 2002]. Their method combines results of Quality Function Deployment and Kano Analysis into a 2-dimensional "needs-opportunity" diagram in which each requirement is plotted as a circle centered at the point located according to that requirement's degree of customer need in one dimension, and proportion of customers who have that need in the other dimension; radius of the circle indicates a measure of the Return On Investment that requirement represents.

We might attempt these approaches in our problem area by plotting computer science areas as requirements (valued in terms of practitioners' needs, and costed in terms of researchers' activities). However, we deal with a much larger number of items (over two hundred leaf nodes in just the D category of the ACM Computing Classification System) compared to the 20 or so requirements on which these authors illustrate their work. We also seem faced with a more open-ended decision space than the equivalent of seeking the optimum set of requirements for a given cost level. For example, we wish to use the information we have gathered to give insights into future areas where research would be beneficial, as well as understand how the identified set of activities meets practitioners' needs.

Note that asking a practitioner to rank the relevance of each researcher's set of activities is not a viable alternative. It assumes overly much knowledge by the practitioner of the research activities, and requires continued update by the practitioner as more researchers are added. It also precludes recognition of the situation that the union of several researchers' activities together meets the practitioner's needs. By a similar argument, researchers cannot rank their relevance to each practitioner's problem. The use of the intermediary taxonomy, familiar to both sides, is key. As mentioned before, we got inspiration for this from JPLer David Tralli's use of DDP to assist activity selection across an entire program of NASA Earth Science Missions [Tralli, 2003].

4.2. Conclusions

The overall aim of this work is to match practitioners' needs to researchers' activities so as to gain insights into the status of entire research programs. These insights should benefit organizations that fund, direct and/or utilize research, researchers who wish to know areas are in need of research and by whom, and practitioners who wish to know what research activities are taking place and who is performing them.

The two key steps of our approach are:

1. Employing a taxonomical classification scheme as intermediary between expressions of need and expressions of activity. This was key to successfully eliciting from practitioners expressions of needs, and from researchers expressions of activities, and thereafter combining them.
2. Inventive use of a risk-centric decision-support tool, which both
   a. suggests a useful analogy in which lack of progress in a given area is a "risk" that adversely impacts attaining practitioners' needs, and which can be mitigated through the contributions of researchers' activities, and
   b. provides the mechanical support needed to handle the volume of information. DDP's mechanisms for information visualization have proven useful for presenting the information in such a way that insights can be made despite the volume of information.

   We were able to use the DDP tool's capabilities for calculation and visualization as is, with the only additional work needed being a small amount of programming to import the data.

One of the assumptions buried within our approach is the definition of how to calculate the contribution of a set of research activities towards meeting a need for advances

within a given area. The formula we presented in section 3.5.1 was motivated by the risk analogy, but is not the only possible way of approaching this calculation. Our feeling is that this problem falls into the category that Ritell termed "*wicked problems*" [Rittel 1972]. Wicked problems have many features, the most important being that no objective measure of success exists. Designing solutions for wicked problems cannot aim to produce some perfectly correct answer since no such definition of *correct* exists. Our approach will be to experiment with several variations of data combination, and find which of the conclusions we extract from the resulting data remain stable across many/all of those variations.

The status of our work is that the preliminary data gathered from 9 researchers and 19 practitioners has been successfully imported into DDP in the manner described and illustrated (albeit with deliberate skewing to avoid disclosure of sensitive information) herein. A second round of data gathering is underway, and DDP is to be used to help gain insight into the aggregate of that data.

Future work will be to inject this capability into the research planning and management processes. The hope is that armed with the kind of information that this approach reveals, research program managers will be better able to match their programs to the emerging needs of long-lived projects. The extension of this approach to study *trends* of research and application is also an area of interest.

5. 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, at West Virginia University under a contract with the National Aeronautics and Space Administration, and at NASA's Independent Verification & Validation Facility in West Virginia. 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, the Jet Propulsion Laboratory, California Institute of Technology, NASA, or West Virginia University.

DDP technology has been developed with NASA's Software IV&V, Code R and Code Q support, managed in part through JPL's Assurance Technology Program Office. The DDP project is led by Dr. Steve Cornford at JPL.

6. References

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

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

[Cornford et al, 2001] S.L. Cornford, M.S. Feather & K.A. Hicks. DDP – a tool for life-cycle risk management. Proceedings *IEEE Aerospace Conference, 2001*, Big Sky, MT Volume 1, pp. 1/441-1/451.

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

[Davis & Hickey, 2002] A.M. Davis & A.M. Hickey. Requirements Researchers: Do We Practice What We Preach? *Requirements Engineering* 7(2), June 2002, pp. 107-111.

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

[Kaindl et al, 2002] H. Kaindl, S. Brinkkemper, J.A. Bubenko Jr, B. Farbey, S.J. Greenspan, C.L. Heitmeyer, J.C.S. do P. Leite, N.R. Mead, J. Mylopoulos & J. Siddiqi. Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda. *Requirements Engineering* 7(3), Oct. 2001, pp. 113-123.

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

[Kulik & Macdonald, 2002] P. Kulik & D. Macdonald. Bridging Gaps in Customer Requirements. *5th European Project Management Conference, PMI Europe 2002*, Cannes, France, Jun 2002.

[NASA-OSMA-SAS 2001] *The 1st Annual Office of Safety and Mission Assurance Software Assurance Symposium*, Sept. 2001. http://www.ivv.nasa.gov/business/research/osmasas/conclusion.htm

[NASA-OSMA-SAS 2002] *The 2nd Annual Office of Safety and Mission Assurance Software Assurance Symposium*, Sept. 2002. http://www.ivv.nasa.gov/business/research/osmasas/conclusion2002.htm

[Rittel, 1972] H.W.J. Rittel. Second generation design methods. *Design Methods Group 5th Anniversary Report*: DMG Occasional Paper, 1, 5-10. Also in N. Cross (Ed.) *Developments in design methodology*, (pp. 317-327). Chichester: Wiley & Sons, 1984

[Rogers, 1983] E. Rogers. *Diffusion of Innovation*. The Free Press, New York, 1983.

[Tralli, 2003] D.M.Tralli. Programmatic Risk Balancing. To appear in the *IEEE Aerospace Conference*, Big Sky MT, 2003.

[Zelkowitz, 1996] M.V. Zelkowitz. Software engineering technology infusion within NASA. *IEEE Transactions on Engineering Management*, 43(3), Aug. 1996, pp. 250-261.