Identifying Fruitful Connections Between and Among Researchers and Practitioners

Martin S. Feather
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, California, USA
Martin.S.Feather@Jpl.Nasa.Gov

Tim Menzies
Lane Dept. of Computer Science &
Electrical. Eng., West Virginia Univ.
Morgantown, West Virginia, USA
tim@menzies.com

Judith R. Connelly
NASA IV&V Facility.
Fairmont, West Virginia, USA
Judith.R.Connelly@nasa.gov

Abstract
Many organizations look to research to yield new and improved products and practices. Connecting practitioners who have the need for research results to the researchers producing those results is important to guiding research and utilizing its results. Likewise, connecting researchers working on related topics to one another, and connecting practitioners with related needs to one another, is important to establishing communities of shared interests. We present an approach that helps identify fruitful such connections.

INTRODUCTION
Many technologically driven organizations look to research to improve their products and the practices by which those products are produced. They wish to reap the benefits that derive from successful technology transfer (flow of ideas from research to practice) and technology infusion (adoption and use of research results by specific organizations).

Our work for NASA gives us some experience of these phenomena. We note that there is continued concern about the low rate at which software technology transfer and infusion takes place: almost a decade ago, [8] reported on a study of impediments to software engineering technology infusion within NASA. This in turn references work from a decade earlier [7]. Many of the observations and insights therein remain valid today. We are currently participating in an effort led by NASA's Software Working Group to improve the infusion of software engineering technology. More generally, these same concerns have recently risen to prominence within the Requirements Engineering community: for example, [6] "...summarises, clarifies and extends..." two conferences' panel discussions on this topic.

One of the impediments to successful infusion is knowing who needs what, and who is working on what. The lack of such knowledge commonly leads to unfulfilled needs, unused research, and unnecessary replication of effort among practitioners who, unbeknownst to each other, share similar problems, and researchers who, unbeknownst to each other, share similar objectives. This problem is exacerbated by the growing number and variety of topic areas, and the increase in the number of venues (workshops, conferences, journals, web sites etc) in which research results are reported.

The focus of this paper is on an approach to overcoming some of these problems. The approach is intended to help *identify* fruitful connections between and among practitioners who have need for research results to the researchers producing those results.

APPROACH
The key idea of our approach is to use a taxonomy (hierarchical tree structure) of research areas as the intermediary through which to relate researchers and practitioners. Researchers express their activities in terms of this taxonomy. Practitioners likewise express their needs in terms of this same taxonomy. We use a decision-support tool to record these expressions of activities and needs, amalgamate them, and visualize the resulting information. The purpose is to help identify:

- fruitful connections between researchers and practitioners by matching the researchers' combined activities to the practitioners' combined needs,
- areas of overlap among researchers, i.e., opportunities for collaboration and sharing of results, and similar areas of overlap among practitioners, and
- "gaps", areas of needs which are unfulfilled (or only weakly fulfilled) by existing research, and areas of research for which there is little or no demand.

Taxonomy
Use of a taxonomy as intermediary avoids requiring that researchers and practitioners directly relate their activities and needs to one another. Stating such relationships directly would presume that each researcher understands each and every practitioner problem in order to know whether to relate to it, or conversely, presume that practitioners understands each and every researcher activity. Use of a taxonomy avoids such problems, provided that:
- such a taxonomy exists or can be created,
- 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.

For example, in our software context we have experimented with using the software section of the ACM Computing

Classification System (1998)[1] [1] as our taxonomy. The "D. Software" section of this taxonomy is divided into:

- D.0 General
- D.1 programming techniques
- D.2 software engineering
- D.3 programming languages
- D.4 operating systems
- D.m miscellaneous

These are further subdivided, e.g., "programming languages" decomposes as follows:

- D.3 programming languages
    - D.3.0 general
    - D.3.1 formal definitions and theory
    - D.3.2 language classifications
    - D.3.3 language constructs and features
    - D.3.4 processors
    - D.3.m miscellaneous

These in turn are subdivided to one further level, e.g., the "formal definitions and theory" is subdivided as follows:

- D.3.1 formal definitions and theory
    - semantics
    - syntax

We take advantage of the tree-structure of the taxonomy to allow expressions of needs and activities to be stated in terms of items at any level of the tree structure, not necessarily the bottommost items. For example, a practitioner with needs spanning a whole subtree within the taxonomy (e.g., "programming languages") can simply indicate an interest in that subtree without the need to be more specific.

In our approach we require that these expressions of activity/need be quantitatively weighted to reflect the relative strength of activity/need (e.g., a researcher active in several topic areas, but to different degrees), and to reflect the magnitude of the activity/need (e.g., one research program may be twice the magnitude of another).

Decision Support Tool

We use a decision-support tool to record practitioners expressions of needs and researchers expressions of activities, to amalgamate them, and to generate visualizations of the combined information.

The tool we use is one we have been developing and applying for risk-informed decision-making. In our JPL and NASA setting this tool is being applied to risk management of spacecraft and spacecraft technologies in their early phases of development. For historical reasons the tool is called "Defect Detection and Prevention (DDP)" – an overview is in [3], and a more extensive description in [4].

Its applicability to this paper's aim stems from the an inventive analogy between DDP's risk-informed decision making, and this paper's aim of matching practitioner needs to researcher activities. We give a brief overview of this – for a more thorough description, see [5].

- DDP: Objectives, Risks and Mitigations – The DDP tool treats "risks" are items that, should they occur, detract from attainment of mission "objectives". Risks can be reduced by application of "mitigations". The primary purpose of DDP is to understand the relationships between objectives, risks and mitigations so as to be able to perform risk-informed decision making (e.g., determine a cost-effective selection of risk-reducing mitigations).
- This paper: Practitioners, Areas and Researchers – practitioners need progress to be made in some areas of computer science. These needs can be met by (successful) research in those research areas. The primary purpose of this work is to understand the relationships between practitioners needs, research areas, and research activities so as to be able to identify fruitful connections among them (e.g., identify practitioners who share the same needs, identify under-fulfilled areas of need worthy of investigation).
- Analogy: practitioners are represented as DDP internal "objectives", research areas are represented as DDP "risks" (this is the most inventive part of the analogy), and researchers are represented as DDP "mitigations". For understandability, we will use the terminology "practitioners", "areas" and "researchers" henceforth, not the names internal to the DDP tool.

When a practitioner indicates a need, this indication is made with respect to the taxonomy of research areas. If a subtree of areas is indicated, the need is automatically subdivided equally among the elements of that subtree (and if that subtree decomposes further, the subdivision process continues). For example, in our software context, if a practitioner indicates a need for research in the ACM Computing Classification System area of "formal definitions and theory", this is automatically subdivided equally among the elements of that subtree, namely "semantics" and "syntax".

Researchers' expressions of activity are treated similarly, subdividing them when they are stated in terms of subtrees.

The net result is that expressions of needs and activities are automatically percolated down to the bottommost "leaf" items of the taxonomy. The more expressions of need that percolate down to a given item, the more total demand there is for research in the area that item represents. The more expressions of activity that percolate down to a given item, the more effectively needs in that area are expected to be met by research in the area that item represents.

It is necessary to make some assumptions about the way quantitative expressions of needs and activities combine. Motivated by our analogy with risk, we use the combination scheme built into our DDP tool. If this were deemed inappropriate, DDP's internal calculation rules would need to be adjusted accordingly, but the overall mechanism would continue to apply. Briefly, our scheme assumes that:

- Need combines *additively* to determine total "demand". For example, if two practitioners express need for the same leaf item of the taxonomy of research areas, then the demand for progress in that area is the sum of those needs.
- Research in the same area combines in such a way as to exhibit a law of "diminishing returns". E.g., doubling the amount of research in an area leads to *some* improvement in how well the needs in that area are met, but the improvement is less than double. We face this same issue in our tool's treatment of risk, so for our current investigations have simply adopted the same scheme we use there. Briefly, it acts *percentage-wise* to determine satisfaction of need. For example, if two researchers express activities each of which is expected to solve 10% of the need for progress in the same leaf item of the taxonomy of research areas, then the combined progress is calculated as follows: 10% of the need is fulfilled by one researcher, leaving 90% unfulfilled; 10% of that 90% (i.e., 9%) is fulfilled by the other researcher, leaving 81% unfulfilled overall. Thus the net combination of both researchers' activities is the equivalent of 81% effective research.

Any approach that aims to evaluate the combined effect of multiple research efforts faces this same decision. It remains to be seen whether it is appropriate to inherit for this purpose the combination rule originally selected for risk informed decision making. For the time being, we are pursuing this route. It is important to note that whatever the scheme, the overall aim is to provide insights which *help* decision makers, not to automate decision making.

PILOT STUDIES

We have performed two pilot studies to investigate the approach. In one, expressions of needs were gathered from 9 NASA practitioners in the area of software verification and validation, and expressions of research activities were gathered from 19 NASA funded software assurance researchers. In another, expressions of research activities were gathered from 19 attendees of a specialist group in the area of software requirements engineering.

Linkage Visualization

An example visualization generated by the decision support tool is shown in Figure 1. The upper half plots expressions of needs gathered from 9 NASA practitioners against the almost 200 leaf items of the "software" portion of the ACM Computing Classification System of computer science. The 9 tiny squares in the row at the top represent each of the practitioners, the 198 tiny squares in the row along the middle represent each of the leaf items of the research taxonomy, and the 19 tiny squares in the row at the bottom represent each of the researchers. A line links a practitioner and an area if that practitioner has expressed need for research in that area (having used the scheme described in the previous section to percolate such expressions of need down to the leaf items of the taxonomy). A line links a researcher and an area if that researcher has expressed activity in that area (again, percolating such expressions down to the leaf items).

For example, the practitioner represented by the leftmost square has expressed need for research in a small number of areas, as seen by the six distinct lines emanating from that square. In contrast, the next-to-the-left practitioner's square

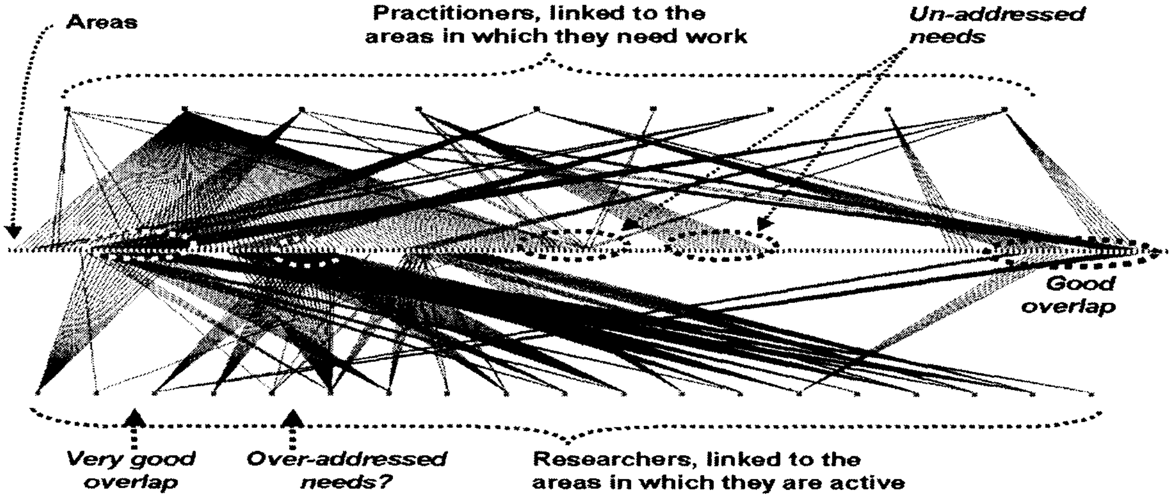

Figure 1. Visualization of needs-areas-activities links

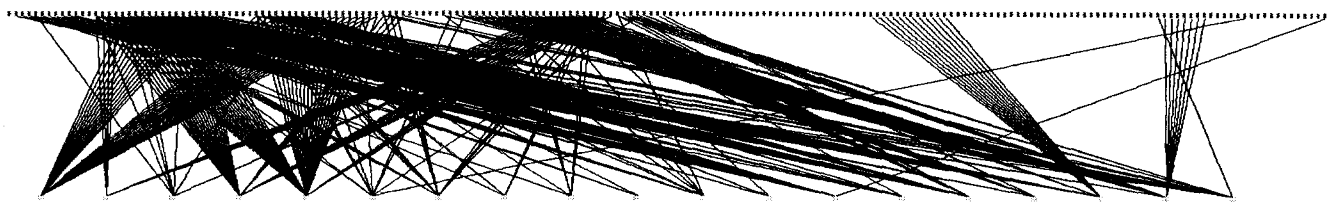

Figure 2. Visualization of areas-activities links for a specialized group of researchers

is linked to almost half of the areas (corresponding to the entire sub-topic "software engineering")!

This visualization allows the immediate visual discernment of several phenomena:

- Where lines from multiple practitioners and multiple researchers converge, this indicates very good overlap between needs and activities. Prominent examples of such are highlighted within the leftmost dotted-ellipse (note: the ellipses are manual annotations to the diagram for the purposes of exposition).
- Overlaps involving fewer practitioners and researchers is highlighted towards the right side of the diagram.
- Where lines from many researchers but few practitioners converge, this indicates practitioner needs that *might* be over-addressed by the researchers (we add the qualifier "might" because it is possible that this is an extremely difficult area that requires a large body of research work to solve).
- Where lines from practitioners are not met by lines from any researchers, this indicates practitioner needs that are un-addressed. Two such areas are highlighted, the left one in particular represents the convergence of several practitioners' expressions of needs, so is particularly striking in this regard.

Please note that our data is drawn from expressions of needs and activities as stated by *some* practitioners and researchers, and thus is by no means a complete representation of status. The insights we draw above are for illustration only!

A similar plot, of just activities as expressed by 19 researchers present at the meeting of a specialist group in the area of software requirements engineering, is shown in Figure 2. (Note: these are *not* the same 19 researchers as were queried for data shown in Figure 1). Since this is a specialist group, it is to be expected that there is clustering of common interests, as indicated by the convergence of multiple researchers' links on the same areas. These match what would be expected from the nature of the specialist group. This suggests the feasibility of using this kind of data gathering and visualization to identify the following:

- Researchers working in the same areas – useful to know to guide formation of communities of shared interests.
- Researchers working in complementary areas.
- Generalists vs. specialists – the former might find it useful to identify the latter to get detailed help, while the latter might benefit from the former's broader contextual knowledge.
- Areas of (non)popularity. These might stem from intention, or could be accidental.

Quantitative Visualization

The linkage visualization indicates only that there is *some* need or activity in a given area. To present quantitative information, namely *how much* need a practitioner has expressed, or activity a researcher has expressed, we employ other visualizations provided by our decision-support tool.

Figure 3 shows a plot where the area of each rectangle is proportional to total demand for research in that area. As discussed earlier, total demand is computed by simply adding together the quantitative expressions of needs as stated by practitioners. This visualization is an instance of one of the "Ordered TreeMap" displays described in [2]. Our tool is capable of generating this plot down to the bottommost level of 198 areas of the ACM Computing Classification System for the software category. In the interests of legibility within the space available, we show here the plot expanded to the level one above that bottom-most level.

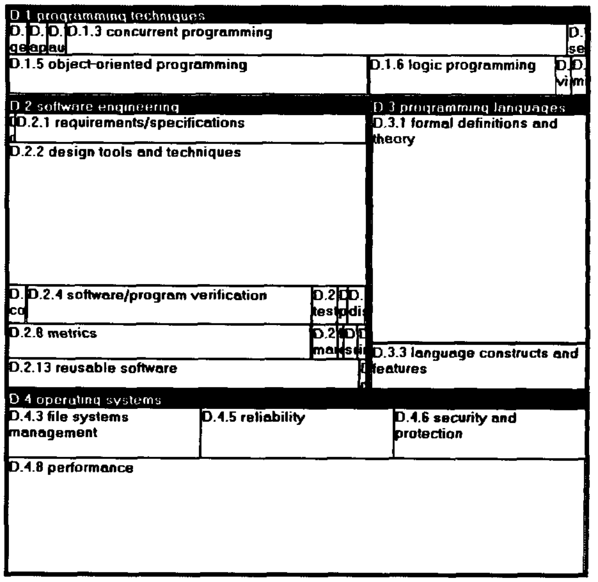

Figure 3. TreeMap visualization of areas' demand.

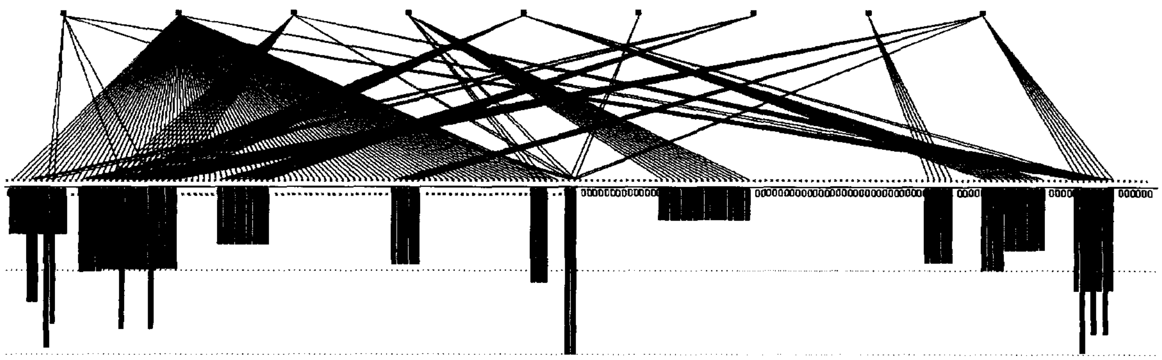

Figure 4. Juxtaposition of practitioner-to-areas links, and magnitudes of total practitioner demand per area

From this visualization we can see at a glance the prominent areas. For example, it is readily apparent that within the "D.3 programming languages category" (middle-right of the diagram) there is large practitioner demand for the area "D.3.1 formal definitions and theory", and somewhat less for "D.3.3 language constructs and features". Areas with only small amounts of demand have correspondingly tiny rectangles, and areas with no demand do not appear at all.

Figure 4 shows another quantitative plot - a bar chart whose bar heights indicate (logarithm of) total practitioner demand. This chart is juxtaposed with the practitioner-to-areas linkage information from the top half of Figure 1. Such combinations allow us to discern areas of greatest demand, and so focus our attention accordingly. Using these same kinds of visualizations we can present quantitative information on the extent to which the current set of research meets the demand in the various areas (not shown here in the interests of space).

CONCLUSIONS

The goal of our work is to help identify fruitful connections between and among researchers and practitioners. In pilot studies of our proposed approach, we have found indications that simple expressions of needs and research can be combined and presented using cogent visualizations in ways that support this goal. The key is use of a suitable taxonomy for the discipline area in question (in our studies, software aspects of computer science). We are interested in exploring the applicability of this approach to other disciplines.

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.

REFERENCES

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

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

[3] 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.

[4] Feather, M.S. & Cornford, S.L.. "Quantitative risk-based requirements reasoning", to appear in *Requirements Engineering* (Springer), published online 25 Feb 2003, DOI 10.1007/s00766-002-0160-y. Available from: http://eis.jpl.nasa.gov/~mfeather/Publications

[5] M.S. Feather, T. Menzies & J.R. Connelly. "Matching Software Practitioner Needs to Researcher Activities", submission to the 2003 Asia Pacific Software Engineering Conference – available from the first author.

[6] 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.

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

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