AC 2009-676: COMPUTING ACROSS CURRICULA: THE VIEW OF INDUSTRY LEADERS

Eric Wiebe, North Carolina State University

Dr. Wiebe is an Associate Professor in the Department of Mathematics, Science, and Technology Education at NC State University. He received his Doctorate in Psychology and has focused much of his research on issues related to the use of technology in the instructional environment. He has also worked on the integration of scientific visualization concepts and techniques into both secondary and post-secondary education. Dr. Wiebe has been a member of ASEE since 1989.

Chia-Lin Ho, North Carolina State University

Dianne Raubenheimer, North Carolina State University

Lisa Bullard, North Carolina State University

Jeff Joines, North Carolina State University

Carolyn Miller, North Carolina State University

George Rouskas, North Carolina State University

Computing Across Curricula: The View of Industry Leaders

Abstract

With the aim of preparing students for pervasive, advanced computing in the workplace, a project funded by the National Science Foundation CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) was initiated in 2007. The multidisciplinary project has two overarching goals: (1) create a computational thinking thread in the engineering curriculum that spans from the freshman to senior years and bridges the divide between freshman year computing and computing in upper-level classes, and (2) enable students to take computing competency to the next level, where they are able to perform high-level computing tasks within the context of a discipline.

The first phase of the project entailed the establishment of an academe-industry community in which stakeholders from a broad range of engineering disciplines convened to discuss the challenges and opportunities inherent in transforming the undergraduate computing education and to identify creative strategies for implementation. To effectively facilitate group communication within the "Computing Across Curricula" (CAC) community, the Delphi method was employed for the systematic knowledge collection and the achievement of consensus among industry leaders. Three critical phases are described in this process: (1) organize an industry panel to identify potential Delphi participants and to generate questions for an open-ended Delphi survey, (2) design, implement, and analyze an open-ended Delphi survey, and (3) design, implement, and analyze a quantifiable Delphi survey. A model of computational capabilities was also derived from the industry panel and is being used to promote a common language of computational capabilities in engineering.

As part of the Delphi process, the project has gathered the first round of feedback from industry leaders, thematically identifying a set of computational capabilities vital for engineering professionals. In the next phase a refined set of computational themes was sent back out to the industry panel for ranking. Results from the industry panel and the current Delphi process will be presented. Implications of the results for a computational thinking thread in the engineering curriculum will be discussed as well as plans for future project activities.

Introduction

Rapidly developing computational technologies are radically reshaping the nature of the workplace [1]. Jobs that consist primarily of routine engineering and computational activities are quickly moving oversees to cheaper labor markets or being completely automated. This and other immense changes in global political and economic dynamics means the 21st century engineer will look very different than their 20th century counterparts [2]. While these changes can be seen as a real threat to the engineering job market, engineers who have learned how to harness computational capabilities for advanced analysis and problem-solving will continue to be in great demand for decades to come.

Therefore, our multidisciplinary National Science Foundation project has a twofold goal to (1) develop a computational thinking thread that spans beyond the freshman year's computing course to all levels of the engineering curricula, and (2) increase students' computational competency in applying appropriate computing approaches during/in the problem solving process.

Developing computationally capable engineers means understanding that changes in the undergraduate engineering curriculum must recognize its context in an educational continuum starting in kindergarten (or before) and ending with a professional engineer prepared for life-long learning. Organizations focused on computing literacies in the K-12 educational arena have been responding to the rapid global changes with new standards and strategies for developing "future-ready" students [3, 4]. Engineering educators have, appropriately, recognized the importance of this preparation prior to starting an undergraduate engineering program [5]. However, true preparation for an engineering career means continuing to build and reinforce these core 21st century skills throughout the undergraduate educational experience [6].

Developing an effective strategy for infusing computational capabilities through an undergraduate engineering program means triangulating data from numerous sources. This project is making use of top-down sources, including reviews of published educational research, as well as bottom-up sources, such as surveying engineering faculty, students, and professional engineers. This presentation will focus on data gathered from professional engineers and then interpreted through a framework derived from published educational research. Our engagement with industry included two primary strategies: a face-to-face daylong workshop and a Delphi process with a second group of industry representatives. The former strategy was used to both gather data in its own right, but also to inform the later Delphi process. Both the experience of engineering education faculty on the project and ongoing reporting in the literature [7, 8] indicates that our strategy would need to include looking at a broad range of engineering disciplines for both commonalities and differences in computational literacy goals

Industry Workshop

With the goal of better identifying potential Delphi participants and refining the open-ended questions of the Delphi survey, a preliminary step of the Delphi process, the face-to-face engagement with our industry leaders, took place on January 25, 2008. The industry panel included thirteen participants and represented companies in the computing, energy, textile and healthcare industries. Participants included senior executives as well as first-line engineering managers and represented five different engineering disciplines. In the workshop, the Delphi process was introduced, and the roles of the industry panel were specified. The industry panel was informed that their work that day would help inform the later Delphi process. To allow for effective and thorough discussion, two subgroups of the participants were created based on their engineering discipline; with one group being more

IT-focused (e.g., Computer Science and Electrical and Computer Engineering) and the other more process oriented (Chemical, Industrial and Systems, Civil, Mechanical, and Textile Engineering). The breakout sessions were facilitated by the project members and involved a discussion of the definitions of computer proficiency and fluency. A small brainstorming activity was facilitated in each subgroup using Affinity Diagrams to answer three potential open-ended questions to be used later in the Delphi process drafted by our project team. The three questions were:

- What proficiencies and fluencies are required for new hires in your company?
- What proficiencies and fluencies do you expect your workers to develop during their first years on the job?
- What new proficiencies and fluencies do you see emerging in the next couple of years in your field?

From the exercise and combination of the results from both subgroups, some common themes emerged as shown in Table 1. The results and feedback from the workshop were utilized to refine the first Delphi survey and also led to the development of a Model of Computational Capabilities.

Table 1 – Common Themes from the Workshop

New hires	After first year on job	Next few years
Specific applications (domain knowledge)	Technological tools	Architecture & technology skills
Problem solving skills (critical thinking)	Systems knowledge	Soft skills (global issues)
Communication skills	Self motivated innovation	Accountability
Knowledge of a programming language	Understanding business needs (value proposition)	Data exploration
Database management skills	Data reporting	

Model of Computational Capabilities

Derived from the discussions in the industry workshop, the need to better define different levels of computational capabilities was identified. The first step was a comprehensive literature review pertaining to computer competency, proficiency, and fluency at the university level. The results of the literature review revealed broad and inconsistent interpretations of the terms competency, proficiency, and fluency with very little material that spoke to the specific needs of the engineering profession. Using the outcomes of the industry panel workshop and literature review, a first draft of a Computational Capabilities Model was completed (Figure 1). This model took the outcomes of the industry panel and shaped it based on a framework of computational fluency derived primarily from an influential National Research Council report [9] and more recent work done by Dougherty and colleagues [10]. The model looks at computational capabilities needed in a problem-solving context—central to both professional engineering practice and, appropriately, engineering education [11]. Basic, relatively stable intellectual capabilities are recognized as essential for problem solving which include the general cognitive abilities necessary for learning and applying declarative and procedural knowledge as well as engaging in the problem-solving process. Technical skills refer to the abilities pertaining to manipulating a specific software tool or programming in a particular language to solve the problem. Two types of specific knowledge also need to be applied to the problem. Conceptual knowledge is higher-level knowledge (i.e., understanding at a more abstract level) of computing systems and languages in general. The application domain knowledge is within the engineering discipline where the problem resides (e.g., polymer synthesis, circuit design, mechanical coupling design).

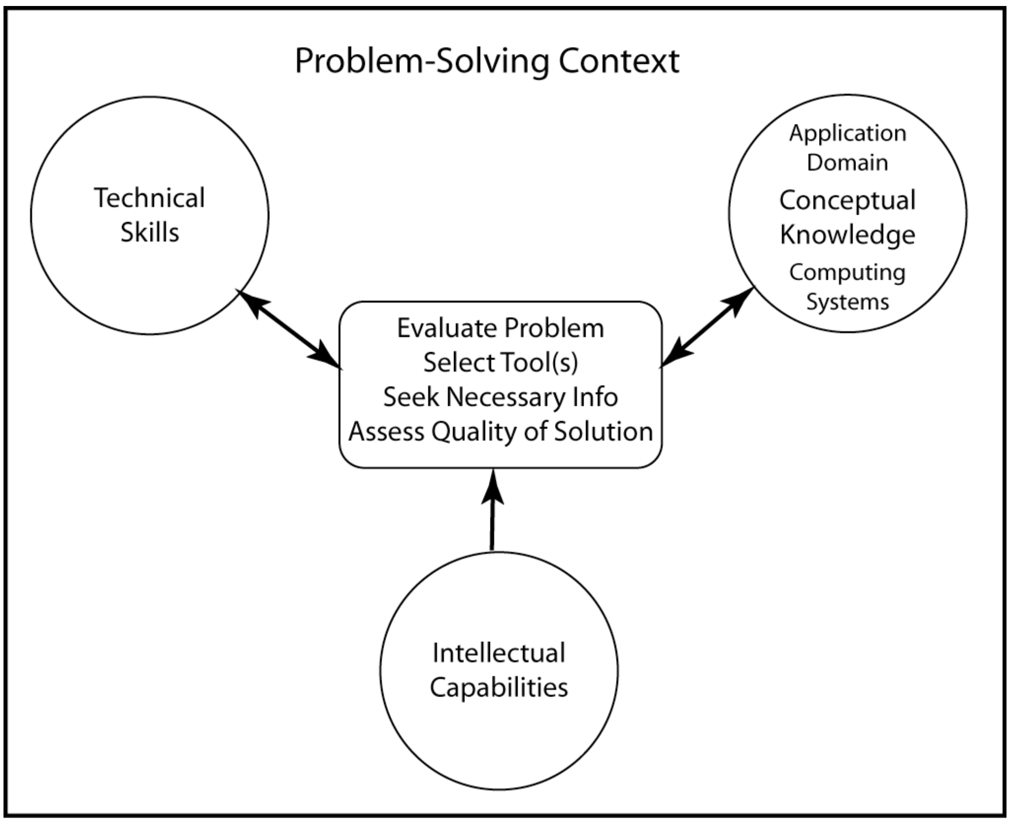

Figure 1 – The Computational Capabilities Model

Using this model, three levels of capability were defined. The goal was to define a curriculum that identifies what general capabilities should be assumed as students leave secondary education and matriculate into an undergraduate engineering program. During their four years in an engineering program, students will continue to develop both general capabilities useful in many areas of their education and specific capabilities to their chosen discipline. One of the hallmarks of the educational context in which these capabilities are applied and developed are problem-solving scenarios that are increasingly ill-defined and complex [12, 13], requiring the Proficiency level of computational capability. It is important to note that the assumption (based on feedback from the industry panel) is that few students will develop capabilities at the fluency level prior to embarking on a professional engineering career. The levels are:

Competency
The individual has technical skill mastery of certain computational tools and/or programming languages. Limits in conceptual knowledge means that they are limited to solving well-defined tasks with specified tools. When faced with a more open-ended or complex problems, limits in conceptual knowledge will mean they will probably not be able to solve the problem.

Proficiency
The individual has some conceptual knowledge of both computing systems and their application domain. When presented with a problem, they are able select the appropriate tools(s), seek the necessary information, and present a solution. The regularly used technical skills are committed to memory and external information resources are not needed in these cases. More complex problems and problems with multiple possible solution paths for which they have to evaluate the quality of the different solution paths will create difficulties for the individual. Overall intellectual capability may be a limiting factor.

Fluency
The individual has extensive knowledge of the technical tools and conceptual aspects of both computer systems and the application domain of their profession. Within their professional area, they are able design and evaluate multiple solution paths to complex problems. They are well versed in general knowledge in the problem space and do not need to refer to external resources for common problems. New computing tools are readily evaluated and integrated into their existing tool set. Limits to problem-solving usually result from moving outside their professional application domain or the bounds of general intellectual capabilities.

Delphi Process
The Delphi process implemented in this project was adapted from McKitrick's (2007) work, as shown in Figure 2. According to Linstone and Turoff (2002), the Delphi technique "may be characterized as a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem (p. 3)." With the goal of systematically obtaining opinions from industry leaders about computing education within engineering disciplines and effectively achieving group consensus, the Delphi process started with the recruitment of participants from industries and the generation of open-ended questions about computational capabilities of engineering graduates. In order to better identify selection criteria of potential Delphi participants and refine the open-ended questions, the face-to-face workshop with our industry leaders took place as a preliminary step.

After administering the first Delphi survey, a qualitative, content analysis was conducted by a group of three subject matter experts (SMEs). As part of the analysis, a coding sheet was first generated to indicate themes among responses. Then all responses were coded by

SMEs; consensus among SMEs emerged. The results of the content analysis will be utilized to design the second Delphi survey for collecting quantitative data about the views of industry leaders on computing education within engineering disciplines. A quantitative analysis will be conducted to identify the consensus among industry partners on the importance of a variety of computational capabilities. If the consensus among industry partners is not achieved, the Delphi process will be repeated. Then the final stage of the process involves reporting consensus results and discussing key discrete opinions.

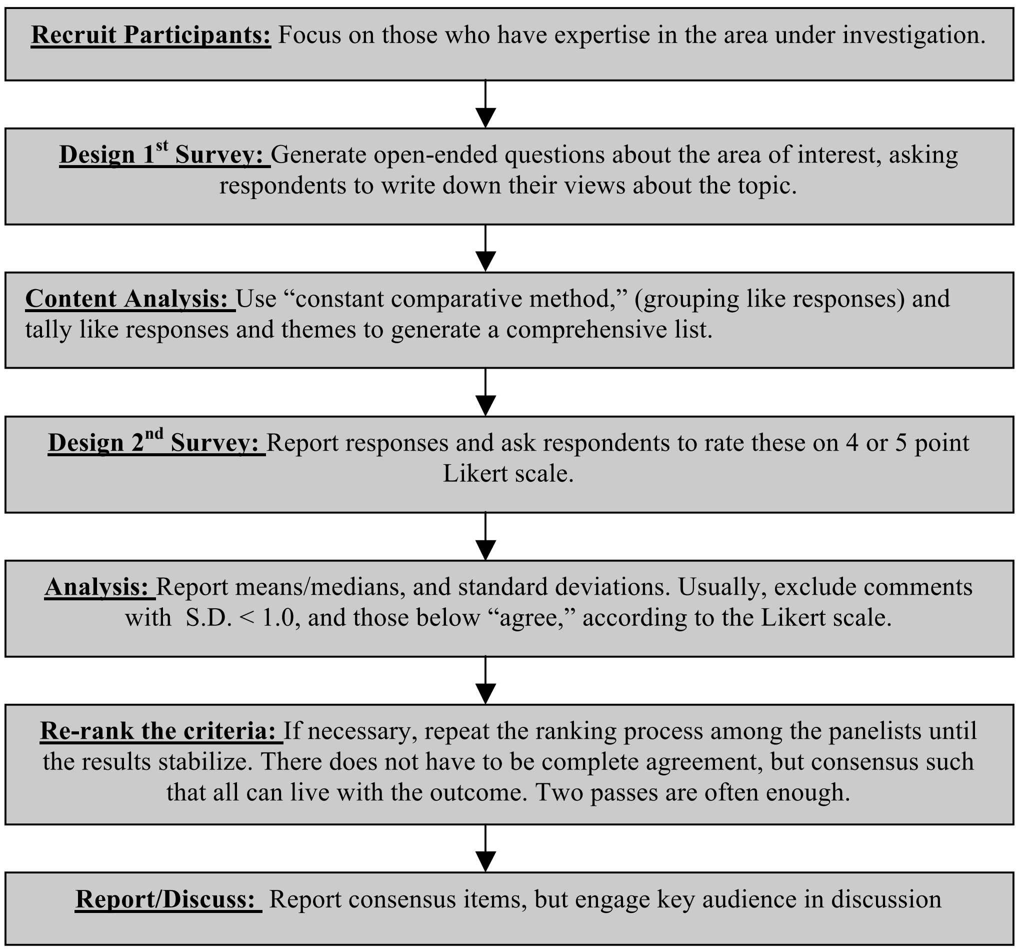

Figure 2 – Delphi Method Procedure

The initial meeting with the industry panel resulted in a set of framing questions that our research team felt would elicit valuable information on a broad set of computational capabilities needed for professional engineers. The questions were crafted to help capture capabilities needed at professional development breakpoints identified by the industry panel. These included capabilities needed at hiring, those that should be developed within the first few years on the job, and new capabilities engineers see coming soon as part of the larger global changes mentioned earlier. Our project team refined the original three questions used in the industry workshop and developed the first Delphi survey with six open-ended questions listed below.

First Delphi Survey Questions

- What computing competencies are required for new technical hires at your company?
- What computing proficiencies do you expect your technical employees to develop during their first few years on the job?
- What new computing skills and processes do you see emerging in the next couple of years in your field?
- Once fluent, what types of problems do you expect your technical employees (with 3-5 years of experience) to solve using computing tools?
- Once fluent, what types of projects do you expect your technical employees (with 3-5 years of experience) to design using computing tools?
- What computing capabilities do you expect technical employees to use to be successful in a global work environment?

Survey participants were recruited via email through networks of our project team and the industry panel attending the workshop. A total of 19 participants represented six different engineering disciplines, with work experience ranging from 1 to 20 years and with the position from first-line engineers to senior managers. The recruitment email included a URL link to the online survey and two attached documents, with one document describing a general Delphi process and the other indicating definitions and examples of the terms *computational competency*, *computational proficiency* and *computational fluency*. These definitions are the same as were developed as part of the Computational Capabilities Model.

The responses of the survey were content analyzed by the group of three SMEs. As the first step of the content analysis, one SME generated a coding list to specify common themes among responses, and the complete agreement on the coding list was achieved among the SME group. This preliminary set of themes was derived based on an initial pass through the responses, data from the industry panel meeting, and the Computational Capabilities Model. Next, all responses were coded by one SME, and the coding results were reviewed and discussed by the SME group. Because of the relatively small number of SMEs in the analysis, complete agreement on the coding results was pursued and achieved in the discussions. As part of this process, some initial codes that were not linked to any statements were discarded and others combined, due to their thematic overlap. The final codes were listed in Table 2.

Table 2 – Finalized Theme Categories from the 1st Delphi Survey

Theme	Code
Analyze & Evaluate Existing Process	AEEP
Ability to Learn & Adaptability	ALA
Ability to Use Simulation Packages	AUSP
Basic Knowledge of Architectures	BKA
Basic Knowledge of Programming	BKP
Basic Operation System	BOS

Communication Tools/organization	COT
Data Analysis Skills	DAS
Driver Concept	DC
Database Fundamentals	DF
Database Management	DM
Forecasting	F
Financial/Interdisciplinary Knowledge	FIK
General: Teamwork, Problem solving (not computing competencies)	G
Industry-specific Tools	IT
Integrated View of Systems/Applications	IVSA
Knowledge of Architectures	KA
Microsoft Office Tools	MOT
None/Not relevant	N
Project Management Applications	PMA
Process Modeling & Design	PMD
Problem-solving & Problem-shooting	PP
Proficiency in Programming Languages	PPL
Proficiency in Simulations	PS
Queries Debugging/Testing	QDT
Security Control	SC
Software Systems Design	SSD
Virtualization	V
Web Programming & Language	WPL
Web Search	WS

Based on notes taken during the creation of these themes, a cluster diagram was created to visualize the larger relationships seen between the thematic categories. Figure 3 represents the resulting five overlapping clusters of themes:

- Computer Science
- Data Analysis
- Design Modeling and Simulation
- Core Individual Work Skills
- Meta-Project Level

Each of these clusters represents both themes that have similarities based on the types of software tools being used and/or the professional context in which they are being used. This professional application context includes both industry-specific tasks and techniques or problem-solving activities that are part of a particular phase of the design process. Themes that were related to each other were put closer together. Where themes had notable linkages to other themes either within or across clusters, vectors were used to connect them.

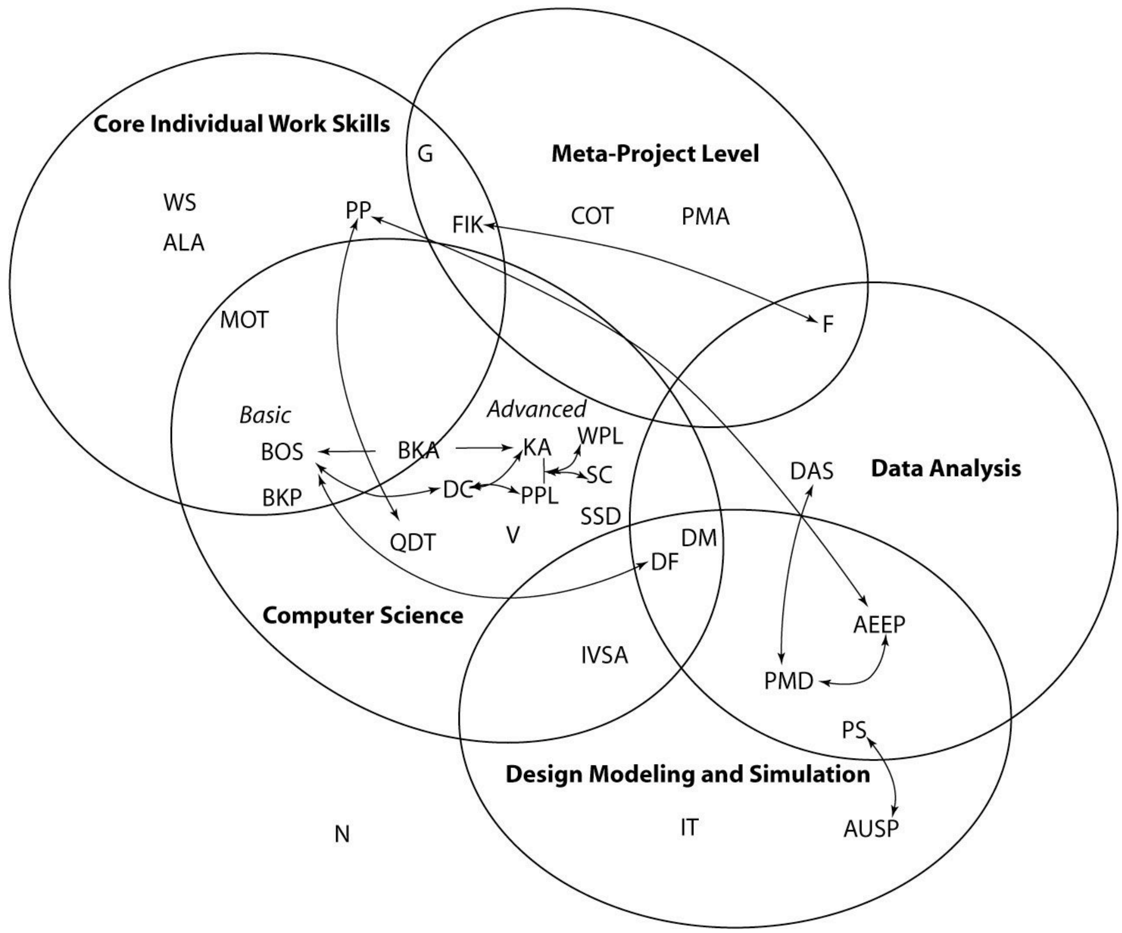

Figure 3 – Theme clustering

Not surprisingly, there was a core set of computing skills that were expected of engineers. The central area of the Computer Science cluster included both basic and advanced level skills in programming and scripting in traditional programming languages and web-based environments. Across the Computer Science cluster, these application and programming skill themes linked at one end to Core Individual Work Skills (e.g., Microsoft Office Applications) to more specialized themes related to engineering problem-solving; more specifically, the management of engineering data (e.g., Database Management, Database Fundamentals).

For some professional areas such as computer science and computer and electrical engineering, much of the day-to-day computational work engineers did stay within the Computer Science cluster. However, other engineering areas had weighted heavily on themes that appeared in both the Design Modeling and Simulation and Data Analysis. By and large, the themes that appeared in Design Modeling and Simulation dealt with computational capabilities that were directly related to core engineering design problem-solving. Data Analysis held themes that were typically related to computational capabilities and that were more indirectly related to the design process, such as managing engineering data used in the decision-making process. While Data Analysis themes may be, at times, only indirectly related to engineering design problem-solving, both the collection and management of data for design is an area of increased emphasis in engineering work and one that students are often much weaker in than professionals [11].

Two of the cluster areas, Core Individual Work Skills and Meta-Project Level contain more general themes related to the professional work of engineers. They relate to the work in a wide variety of engineering professions and all aspects of their day-to-day work. As indicated by their names, Core Individual Work Skills focuses primarily on themes related to computational capabilities applied to individual work while Meta-Project Level themes relate more to capabilities associated with team or project oriented work. Meta-Project Level themes also relate to connecting engineering design work to other aspects of the companies' work. Many of these "soft skills", while not directly related to engineering problem-solving, have long been recognized by engineering educators as key capabilities valued by the engineering profession [14, 15].

Recommendations and Conclusions

The combination of both the bottom-up process of speaking directly to engineering professionals and using frameworks derived from the research literature to shape and interpret our data has been a powerful approach to provide input into the computational capabilities graduates need to succeed as 21st century engineers. The analysis and framing outlined in this paper will be used to guide the development of a second round of statements that the Delphi participants from industry will rate. From this, the Delphi results will be operationalized as a set of strategies the team will be using with our ongoing professional development interventions with engineering faculty. Our goal is to support faculty in revisions of their courses to provide a richer set of computational experiences in the context of engineering problem-solving. To this end, the desire is to have graduates who have the appropriate level of capabilities (competent or proficient) to meet the expectations of their future employers.

Acknowledgement

This work is supported by NSF (CISE # 0722192) as part of CISE Pathways to Revitalized Undergraduate Computing Education program. The project team would also like to extend its sincere thanks to our partners in industry who served on our panels and our CAC fellows who are implementing their innovations in their classrooms.

References

1. Levy, F. and R.J. Murnane, *The new division of labor: How computers are creating the next job market*. 2004, Princeton, NJ: Princeton University Press.
2. Vest, C.M., *Context and Challenge for Twenty-First Century Engineering Education.* Journal of Engineering Education, 2008. **97**(3): p. 235-236.
3. Dede, C., *Determining, Developing and Assessing the Capabilities of "Future-Ready" Students*. 2009, Harvard University: Cambridge, MA.

4. ISTE, International Society for Technology in Education, *The national educational technology standards and performance indicators for students*. 2007, ISTE: Eugene, OR.
5. Gonzales, R.F. and S. Renshaw, *Connected Computing - A Hierarchy of Pre-Engineering Computing Skill Competencies*, in *Annual Meeting of the ASEE*, ASEE, Editor. 2005, ASEE.
6. P21, Partnership for 21st Century Skills. and The College Board, *Are they really ready for work?* 2006, Partnership for 21st Century Skills: Tucson, AZ.
7. Davison, L. and N. Porritt, *Using computers to teach.* Proceedings of the Institution of Civil Engineers-Civil Engineering, 1999. **132**(1): p. 24-30.
8. Kassim, H.O. and R.G. Cadbury, *The place of the computer in chemical engineering education.* Computers & Chemical Engineering, 1996. **20**: p. S1341-S1346.
9. NRC-CITL, National Research Council - Committee on Information Technology Literacy, *Being fluent with information technology*. 1999, Washington, DC: National Academies Press.
10. Dougherty, J.P., et al., *Information Technology Fluency in Practice.* ACM SIGCSE Bulletin, 2003. **35**(2): p. 153-171.
11. Atman, C.J., et al., *Engineering Design Processes: A Comparison of Students and Expert Practitioners.* Journal of Engineering Education, 2007. **96**(4): p. 359-379.
12. Stevens, R., et al., *Becoming an Engineer: Toward a Three Dimensional View of Engineering Learning.* Journal of Engineering Education, 2008. **97**(3): p. 355-368.
13. Strobel, J. and M. Cardella. *Compound Problem Solving: Workplace Lessons for Engineering Education*. in *ASEE Annual Meeting*. 2008. Pittsburgh, PA: ASEE.
14. Katz, S.M., *The Entry-Level Engineer: Problems in Transition from Student to Professional.* Journal of Engineering Education, 1993. **82**: p. 171-174.
15. Kranov, A.A., et al. *A Direct Method for Teaching and Assessing Professional Skills in Engineering Programs*. in *ASEE Annual Meeting*. 2008. Pittsburgh, PA: ASEE.