AC 2008-740: COMPUTING ACROSS CURRICULA

Amy Craig, North Carolina State University

Amy E. Craig is the Coordinator of Student-Owned Computing in the College of Engineering and a doctoral student in the Department of Industrial and Systems Engineering at NC State University. She regularly teaches the Introduction to Engineering and Problem Solving course in the First Year Engineering Program. Her research interests include faculty development and teaching and learning in the engineering disciplines. She received her MIE and BSIE degrees from NC State University. Prior to her return to NC State, she worked as a Cost Engineer in the Personal Computing Division of IBM.

Lisa Bullard, North Carolina State University

Lisa G. Bullard is a Teaching Associate Professor and the Director of Undergraduate Studies in the Department of Chemical and Biomolecular Engineering at NC State University. She received her B.S. in ChE from NC State and her Ph.D. in ChE from Carnegie Mellon, and she served in engineering and management positions within Eastman Chemical Co. from 1991-2000.

Jeff Joines, North Carolina State University

Jeffrey A. Joines is an Associate Professor in the Textile Engineering, Chemistry, and Science Department at NC State University and is currently the Director of the Textile Engineering Program. He received a B.S. in Electrical Engineering and B.S. in Industrial Engineering in 1990, a M.S in Industrial Engineering in 1990, and Ph.D. in Industrial Engineering in 1996 all from NC State University. He received the 1997 Pritsker Doctoral Dissertation Award from the Institute of Industrial Engineers. He was awarded the 2006 NC State University Outstanding Teaching Award. He also serves as the faculty chair of the Student-Owned Computing (SOC) initiative in the College of Engineering. His research interests include evolutionary optimization, object-oriented simulation, simulation-based scheduling and supply chain optimization. He was the Co-Proceedings Editor for the 2000 Winter Simulation Conference and the Program Chair for the 2005 Winter Simulation Conference and acts as the technical coordinator for the conference's management system.

Carolyn Miller, North Carolina State University

Carolyn S. Miller is a Lecturer in the Department of Computer Science at NC State University. She received her M.S. in Computer Science and worked as a Member of the Technical Staff at Bell Telephone Laboratories and a Senior Digital Systems Engineer at General Electric before joining NC State University. Ms. Miller teaches introductory computer science classes and focuses on researching and integrating new teaching techniques into the classroom.

Tom Miller, North Carolina State University

Thomas Kenan Miller, III received the BA degree in Mathematics and Chemistry from the University of North Carolina at Chapel Hill in 1976. He received the MS degree in Biomedical Engineering and Mathematics in 1980, and the PhD in 1982, both from UNC-Chapel Hill. Since 1982 he has been with the Department of Electrical and Computer Engineering at North Carolina State University, where he holds the rank of Professor of Electrical and Computer Engineering, and currently serves as Associate Dean for Distance Education and Information Technologies. In his role as Associate Dean, Dr. Miller is responsible for strategy, deployment, and implementation of the College of Engineering's academic and administrative computing networks and distance education delivery technologies.

In July 2000, Dr. Miller was named Interim Vice Provost for Distance Education and Learning Technology Applications, NC State's newest division within the Office of the Provost, and was then given the appointment of Vice Provost for DELTA in January 2001.

Dianne Raubenheimer, North Carolina State University

Dianne Raubenheimer is Director of Assessment in the College of Engineering and Adjunct Assistant Professor in the Department of Adult and Higher Education in the College of Education at NCSU. She has worked with faculty and administrators in Engineering for three years, and previously in the Science and Education disciplines. She has a background in teacher education, curriculum development and evaluation and has worked as an education consultant for a number of organizations in the USA and South Africa conducting program evaluations. She received a Ph.D. in Educational Leadership and Organizational Development (Higher Education) from the University of Louisville and has M.Ed, M.Sc, B.Sc (Hons) degrees and a postgraduate Diploma in Adult Education from the University of Natal, Durban, South Africa.

George Rouskas, North Carolina State University

George N. Rouskas is a Professor of Computer Science at NC State University. He received the Ph.D. and M.S. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1994 and 1991, respectively, and his undergraduate degree in Computer Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989. He received the 2004 ALCOA Foundation Engineering Research Achievement Award, and the 2003 NC State Alumni Outstanding Research Award. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award and of the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech. He was inducted in the NC State University Academy of Outstanding Teachers in 2004. He founded and serves as co-editor-in-chief of Optical Switching and Networking (OSN), an Elsevier journal, and he has organized several conferences and workshops.

Larry Silverberg, North Carolina State University

Larry M. Silverberg is a professor in the Mechanical and Aerospace Engineering Department and currently serves as Associate Head. He received a BS in Engineering, Science and Mechanics in 1980, an MS in Engineering Mechanics in 1981 and a PhD in Engineering Mechanics in 1983 from Virginia Tech. Among his honors, he is a member of the University Academy of Outstanding Teachers, a recipient of the College of Engineering Outstanding Teacher Award in 1995, received the NASA Outstanding Contribution Award in 1994 and the Alcoa Foundation Engineering Research Achievement Award in 1993. Since coming to NC State he published about 50 articles in archival journals, 2 books, 2 patents, and developed 2 graduate courses. His theoretic research lies in the areas of field theory and control of dynamical systems. On the more applied side, he developed the OE-6 Orbiter Ejector which flew on 3 shuttle missions (STS 47 in 1992, STS 53 and STS 60 in 1994), developed light-regulating insulated glass and developed trans-permanent magnetic actuation technology for ultra-low power devices (HVAC dampers, solenoids).

Eric Wiebe, North Carolina State University

Eric N. 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.

Computing Across Curricula

Abstract

Today's somewhat disjointed approach to discipline-specific computing and generic computer literacy does not accurately mirror the knowledge, skills and abilities needed by the engineer of the future global workplace. Computing in the workplace is pervasive and involves many complex tools, many approaches to problem solving, strategic decision making and synthesis. Knowledge, comprehension and application are no longer enough for one to be labeled highly competent in computing. A successful engineer will need a mastery of computing applicable to the higher level cognitive skills of analysis, synthesis and evaluation, as well. To prepare students for pervasive, advanced computing in the workplace, we must begin to think in terms of pervasive, advanced computing in their education. This premise served as the basis for a project funded by the National Science Foundation CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) initiative in 2007. The project is being carried out by the authors comprised of a multidisciplinary team of faculty from six departments in the College of Engineering and one from the College of Education at NC State University.

The 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 entails the establishment of an academe-industry community in which stakeholders from a broad range of disciplines will convene to discuss the challenges and opportunities inherent in transforming the undergraduate computing education, and to identify and implement creative strategies to do so. The "Computing Across Curricula" (CAC) community includes involvement from a number of local industry leaders and is modeled after the nationally recognized NC State Campus Writing and Speaking Program[1] that promotes writing across the curriculum.

To ensure that the goals and objectives of the project are met successfully, a detailed assessment process is an integral part of the ongoing activities. One research and assessment method being employed is the Delphi method. This formal communication process will be used to gather consensus among faculty, industry leaders and students in an effort to identify sets of computational skills vital for engineering professionals. Descriptions and reflections of the first year of project activities will be presented as well as plans for future activities. Preliminary assessment data will also be available.

1. Introduction

Two decades ago to say that an engineering graduate was highly competent in computing meant that he or she had mastered the FORTRAN programming language. A decade later it meant, in addition, mastering basic skills in a few key discipline-specific software tools, and a degree of proficiency in ubiquitous productivity applications such as spreadsheets and word processors. An engineering graduate who could do these things was said to be "computer literate." Today, however, this somewhat disjointed approach to discipline-specific computing and generic computer literacy does not accurately mirror the knowledge, skills and abilities needed by the engineer in the global workplace. Computing in the workplace, and of particular interest here the engineering workplace, involves many complex tools, many approaches to problem solving, strategic decision making and synthesis. Can I decompose this problem and understand which components might require me to write a program and which can be handled by a spreadsheet? How do I move information between the two? How will the results be used? Can my dataset be handled with flat files, or does it scale such that I need a relational database? Should I build, buy, or use an open source solution? These are some of the critical questions facing today's engineering graduates in the workplace. To put it in the context of Bloom's taxonomy, knowledge, comprehension and application are no longer enough for one to be labeled highly competent in computing. The successful engineer of the 21st century will need a mastery of computing applicable to the higher level cognitive skills of analysis, synthesis and evaluation, as well. To prepare students for pervasive advanced computing in the workplace, we must begin to think in terms of pervasive, advanced computing in their education.

This premise served as the basis for a project funded by the National Science Foundation CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) initiative in 2007. The project is being carried out by the authors comprised of a multidisciplinary team of faculty from six departments in the College of Engineering and one from the College of Education at NC State University.

1.1 The Computational Thinking Thread

In the words of Dr. Jeannette Wing, "Ubiquitous computing was yesterday's dream that became today's reality; computational thinking is tomorrow's reality".[2] According to Dr. Wing, computational thinking "requires thinking at multiple levels of abstraction", is a fundamental skill everyone should know, and "is a way humans solve problems; it is not trying to get humans to think like computers".[2] Computing has become a prerequisite skill in engineering. As with mathematics, computing impacts the quality of education because it drives instructor expectations with regard to the nature and complexity of the problems that can be tackled. In the absence of a clear sense of computing expectations, engineering education suffers.

Computing is a subject that the student learns incrementally. Computing can be thought of having three educational phases: the *introductory phase*, the *training phase* and the *proficiency phase*. At the freshman level, the student learns a variety of computer programs and platforms; this is the introductory phase. In the sophomore and junior levels, the student gains experience with computing in his/her engineering classes; this is the training phase. By the senior year, the student is expected to have gained a perspective on the strengths and weaknesses of the different languages, platforms and computing technologies and to have become proficient at each. When confronted with a problem to solve, the student learns how to use computing tools in the problem-solving process, for example, how to use computing to manage data, to program and utilize vector operations, and to display information; this is the proficiency phase.

The process by which engineering curricula insure that students are proficient in certain tools is through what are commonly called prerequisites. For example, mathematics prerequisites are quite common in any engineering curriculum. However, computing is different and herein lies one of the challenges. Computing in engineering is a capability, unlike mathematics, that is intended to grow from the freshman year to the senior year and therefore cannot be managed effectively through prerequisites. The introductory phase can be handled by prerequisites, but the training and proficiency phases cannot be managed this way because they are incremental in nature. After the initial freshman introductory phase that often takes place in a stand-alone class, the next two phases are often more implicit than explicit and tightly bound to the context of a particular class. Students are no longer taking a separate course learning generic computing skills that are prerequisites for particular intermediate or advanced engineering courses. Instead, the computing experience is interwoven through multiple courses over the next three years of a student's academic career, such that the latter two phases may involve over half of the core classes in a major. Yet, students are expected to extract their computing experiences from one course and apply it to problems in others, building a broad, flexible foundation of computational thinking.

How can a curriculum manage the education of computing at the training phase? How do we know what students need to know in this phase? How can individual classes and curricula be developed effectively so instructors have a more accurate sense of the students' computing capabilities? How can computing be assessed to insure continued and improved quality? Engineering curricula face challenges in establishing, maintaining and assessing what we call the "computational thinking thread." Without a clear understanding of its strengths and weaknesses, the level of student proficiency at the different stages is vague, and we hypothesize that instructor expectations are lowered and the quality of the engineering programs does not reach its true potential.

The rest of the paper describes our approach to addressing these important questions, and is organized as follows. Section 2 describes our vision for building an academe-industry community of stakeholders. In Sections 3 and 4 we present our implementation and assessment plans, respectively, as well as our progress and activities to date. We provide some concluding remarks in Section 5.

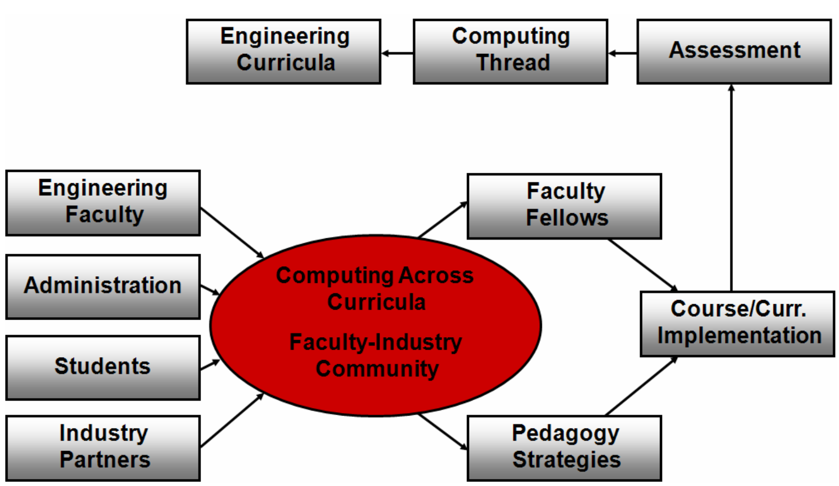
2. The Computing Across Curricula (CAC) Community

In an effort to address the challenges associated with the "computational thinking thread" and to ensure relevance of the engineering curricula, we are working to establish an academe-industry community. Faculty learning communities are nothing new to academia. The goal of the CAC community is to build upon this proven method of faculty collaboration by extending it to include representation from local companies. Although university-only communities can be an effective means of addressing many issues pertinent to education, we feel that considering the industry point of view is crucial to our community. Indeed, data from a recent alumni survey indicated that many students perceive a gap between what they learned in college and their actual experiences in the workforce. Furthermore, several students suggested that faculty interaction with industry on an ongoing basis would be valuable in aligning the curriculum with industry needs. Therefore, a key focus of our activities will be to seek and encourage a variety of key computing stakeholders outside of the university to participate. Our vision is for the CAC community to serve as a vehicle for faculty to partner with local industry leaders in an effort to initiate meaningful channels for dialogue and ideas to flow from industry to the university.

We also recognize that without broad and sustained faculty involvement, the community is likely to fade quickly without coming close to realizing our vision. Consequently, encouraging and motivating faculty outside the initial team of faculty to participate in the community is at the core of our strategy. After the initial planning phase and pilot, each semester we will invite a different group of faculty spanning all engineering disciplines to become involved. These faculty members will participate in a set of structured activities that include attending regular community meetings and implementing relevant enhancements in their respective courses. At the end of the semester, participating faculty will be designated "faculty fellows" and will be invited to share their experiences at future community gatherings. Studies[3] have shown that the existence of even a small number of faculty within each department with expertise on integrating technology and innovative pedagogy is sufficient to drive wide-spread adoption of similar practices across a university. Therefore, we expect the faculty fellows to become the driving force behind the eventual transformation of the engineering curriculum.

Figure 1 illustrates the role of the CAC community in bringing together an interdisciplinary mix of academic and industry stakeholders to: (1) identify shortcomings in computing education within engineering disciplines, taking into account industry needs, and codify the high-order computational thinking outcomes our graduates should possess for problem solving and decision making in the workplace; (2) operationalize these outcomes as pedagogical strategies that address existing shortcomings and needs; and (3) put in place an evaluative structure to determine if the pedagogical strategies have been effective at addressing the desired outcomes identified with our industrial partners. We emphasize that this project is only a first step towards identifying the elements of the computational thinking thread. Transforming the entire engineering curriculum by implementing a common computational thinking thread across the disciplines is outside the scope of this project, but will be considered as a follow-up activity.

Figure 1: Computing Across Curricula Community



2.1 Objectives and Outcomes

Specific objectives of the project are to:

- *Assemble a community of stakeholders from both academia and industry.*
  The community will engage in a variety of activities designed to promote interactions among the stakeholders, and will hold regular sessions in the form of meetings, seminars, workshops and retreats.
- *Identify effective strategies for integrating computing into curricula across multiple disciplines.*
  Faculty, often working as a pair with an industry representative, will develop innovative pedagogy and learning methods using technology, and will design high-level projects and innovative assignments that emphasize a data-centric approach to engineering education.
- *Develop faculty from across the curricula as leaders in computing education.*
  Faculty from across the curricula who participate in the community for one semester will "graduate" to faculty fellow status, and will serve as liaisons between the community and their respective departments, promoting and encouraging the adoption of new methods and technologies in classrooms.
- *Increase coordination and collaboration amongst faculty from various disciplines.*
  The community will identify computing abilities in support of high-level cognitive skills of analysis, synthesis and evaluation that are necessary for tackling complex engineering problems that cut across disciplines.
- *Take initial steps towards introducing pervasive computing in the engineering curriculum.*
  As the community stabilizes, we will undertake broader and more systemic activities aimed at identifying common intra- and inter-disciplinary threads in computing applications, and build on these to begin the process of identifying a common framework that points towards the higher-order abilities our students should possess. This framework will be used to help design instructional approaches that facilitate a student's ability to carry forward and build on his or her abilities as they progress from course to course and into the workforce.
- *Nurture future engineering leaders.*
  The course and curriculum changes effected as a result of the community's activities will provide our graduates with the computing skills necessary to lead successful engineering careers.

3. Implementation Plan

The community activities and implementation plan described in this section will be carried out by the authors consisting of an interdisciplinary team of faculty from six departments in the College of Engineering and one from the College of Education at NC State University, including the director of assessment from the College of Engineering and a project coordinator. As the project progresses and our understanding of the issues involved matures, we plan to invite faculty members in the humanities, social sciences and other disciplines to participate as faculty fellows. Eventually, our goal is to expand the scope of the CAC community campus-wide and to seek partnerships from other universities.

3.1 Community Formation, Organization and Management

During the first year of the project our focus has been on planning and pilot activities. The faculty members involved formed the basis for the multidisciplinary community, and are serving as the inaugural faculty fellows.

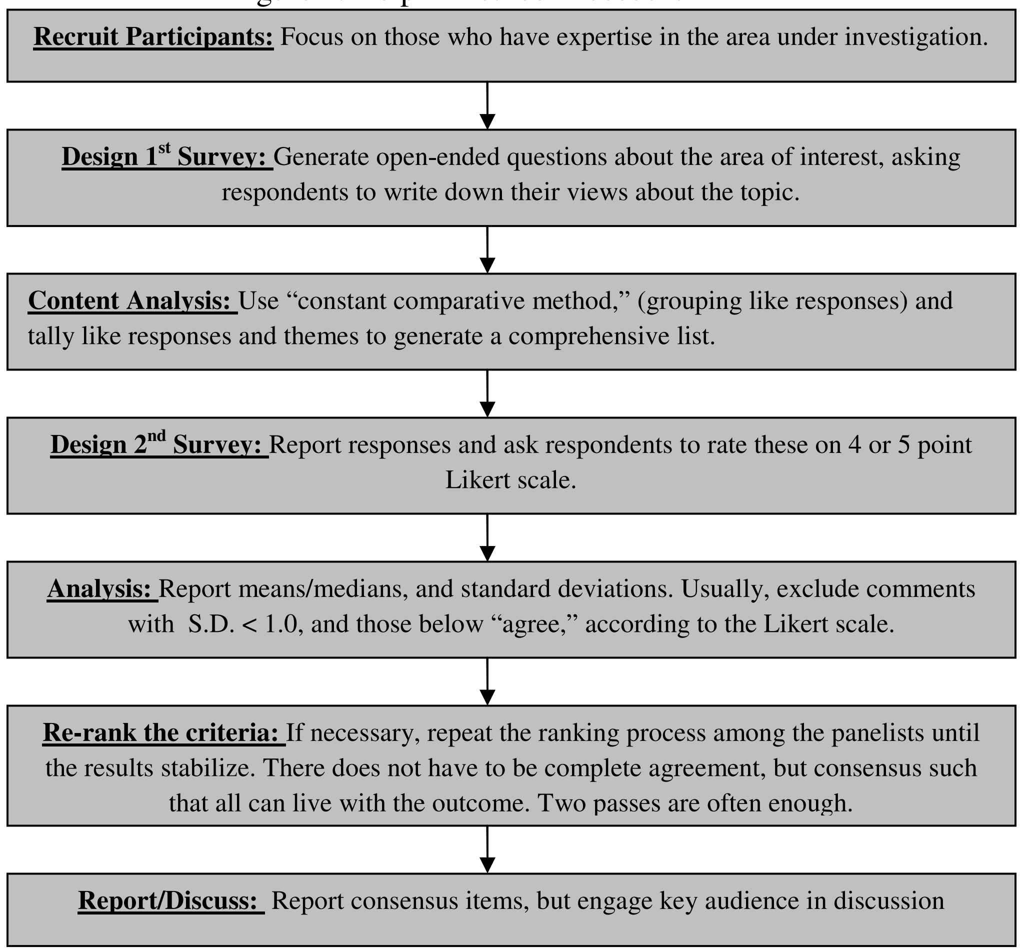
3.1.1 Planning: In the first semester of the project, Fall 2007, the team engaged in the following activities:

- Obtained the use of a project management website and posted and organized all documents, meeting minutes and project milestones;
- Reviewed literature pertaining to computer literacy and fluency at the university level;
- Explored the use of the Delphi[4] method and decided upon its use in obtaining feedback from industry partners regarding their expectations of new hires with respect to computing skills;
- Created and submitted a baseline survey to faculty in various engineering departments to learn about their use of computing in their courses;
- Met with the director of the nationally recognized NC State Campus Writing and Speaking Program for which the CAC community was modeled;
- Developed plans for an industry workshop to be held in January 2008 and contacted 20 industry partners to solicit their participation; and
- Formalized plans for the community pilot semester and explored the use of action research.

Figure 2 shows a schematic of the Delphi method procedure. According to Linstone and Turoff, "Delphi 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".[4] In this case, the "complex problem" is infusing computing competencies into engineering courses. The director of assessment will lead the process, which will be initiated at the industry workshop by having the industry partners assist us in refining the open-ended questions associated with the procedure.

The meeting with the Campus Writing and Speaking Program director was particularly enlightening as he offered guidance as to how our community should be structured. He discussed how his program started, the structure of their workshops and seminars, their processes for faculty recruitment and recognition, the role of the graduate student assistants and program assessment.

Figure 2: Delphi Method Procedure[5]



3.1.2 Industry Workshop: Our first face-to-face engagement with our industry partners took place on January 25, 2008. The primary goal of the workshop was to open meaningful channels of dialogue with our industry partners to learn about their views on computing education within engineering disciplines. The partners were encouraged to identify any shortcomings in their own engineering education and to share any proposed improvement strategies for ensuring that our graduates possess appropriate skills for problem solving in the workplace. The faculty defined the following four outcomes for the industry partners attending the workshop which were evaluated at the end of the session via a feedback form distributed to each participant:

- Assist in defining computer fluency and proficiency in the context of engineering.
- Identify characteristics of individuals that should participate in the Delphi process.
- Generate the open-ended questions for the Delphi process.
- Discuss expectations of new hires in their organizations with regard to computing skills.

Thirteen participants attended the workshop, representing 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.

The workshop commenced with opening remarks from the Vice-Provost followed by an introduction to the project provided by the lead PI. The director of assessment presented a brief introduction to the Delphi method and explained how we would be testing open-ended questions during the workshop. Next, the participants were broken into two groups based on their engineering discipline, with one group being more IT-focused (CSC and ECE) and the other more process oriented (ChemE, ISE and TE). The breakout sessions were facilitated by the faculty fellows and began with a discussion of what computer proficiency and fluency means in the context of higher education. Prior to the workshop, the faculty fellows drafted three questions that could potentially serve as the open-ended Delphi questions provided that the questions were well received and understood by the industry participants. The 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?

The facilitators wrote one of the statements “New hires”, “After first year on job” and “Next couple of years” on each of three chalkboards in the room to coincide with each of the three aforementioned questions. Then an adaption of an affinity diagram procedure[6] ensued where the participants were given time to record their ideas about each question on individual post-it notes and to place them on the appropriate chalkboard. Pairs of participants were asked to organize the post-its on their respective chalkboards to capture common or emerging themes. Some prevalent themes are shown in Table 1:

Table 1: Select Workshop Notes

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	

Following the breakout sessions, the groups reconvened to present a summary of their break-out session discussions to the entire group. The faculty fellows summarized the common themes that emerged from the discussions and the workshop concluded.

A thorough formal analysis of the participants’ responses is underway and more data will be available at the conference presentation. The notes and video transcripts from the workshop will be used to create and refine the open-ended questions that will be used in the first survey of the Delphi process.

3.1.3 Community Pilot: In the second semester of the project, Spring 2008, the inaugural faculty fellows will pilot the activities of the aforementioned planning phase and make refinements as necessary to the activities proposed. During the pilot, the faculty fellows will attend two 2-hour sessions per month facilitated by the project coordinator. The industry panel, consisting of 4-6 representatives, will be initiated during the pilot semester, perhaps including some of the participants from the industry workshop. Industry involvement is a critical component of the community; the logistics of the representatives' involvement will be coordinated during this phase of the project. The industry panel will convene at least twice during the semester, during a scheduled community session. Many community sessions will be structured such that the industry partners will pose topics relating to the current state of computing competencies in the U.S. workforce, and discuss skills characteristic of successful computing professionals in the field. Faculty fellows will use their pedagogical experience to translate the industry inputs into practical assignments, curricular adjustments, etc. This exchange of meaningful dialogue between industry partners and faculty fellows will be an integral part of the community.

During sessions when the industry panel is not in attendance, session formats and topics will vary. Per guidance from the Campus Writing and Speaking Program director, each session will have a theme (for example: Computing & Creativity) which will be presented to the faculty fellows at the beginning of the session, followed by the faculty working on their individual course projects. In preparation for the pilot, each faculty member was asked to pose a research question with regard to computing in their course and they will use the iterative process associated with action research to test out potential strategies. According to Kemmis as cited by Hopkins: *Action research is a form of self-reflective enquiry undertaken by participants in social (including educational) situations in order to improve the rationality and justice of (a) their own social or educational practices, (b) their understanding of these practices, and (c) the situations in which the practices are carried out. It is most rationally empowering when undertaken by participants collaboratively ... sometimes in cooperation with outsiders.*[7] Some examples of the action research projects to be undertaken in Spring 2008 include: having students work in groups to match data structures and algorithms they find on the web, developing an instructional handbook for students featuring tools and computational problems that are specific to chemical engineering, introducing virtual office hours into a circuits course and developing new in-class exercises in a computer programming course.

The planning of the various community sessions will be guided by the following three principles that were discovered during another successful NSF project affiliated with NC State University and that have been reinforced by several other studies:

1. *Faculty development programs need to emphasize disciplinary relevance. Perceived relevance is perhaps the single most important feature of faculty development programs that induces engineers to sign up for them and to take them seriously. In workshops and seminars, include discipline specific examples of recommended teaching strategies.*
2. *Faculty development programs need to keep it practical because a critical characteristic of successful engineering programs is their perceived practicality. Most engineers who attend teaching workshops are not seeking philosophical discussions about the nature of learning; they just want to know what they can do next Monday to make their classes work better. Some material from educational and cognitive psychology (especially*

research data) is essential, but it should be brought in to support the practical ideas that constitute the bulk of the workshop rather than being an end in itself.

3. *Faculty development programs need to include both disciplinary and pedagogical expertise on workshop facilitation teams. A workshop co-facilitator with an engineering background can easily construct practical examples and exercises with technical content. Many engineering faculty members who come to our workshops do so because they know that one of the facilitators is one of them, and that one goes out of his way to reinforce that notion early in the workshop, injecting terms like partial differential equations and entropy whenever he can shoehorn them into the discussion. Once the participants hear those magic words they are more engaged.*[8]

In general, a broad range of projects and initiatives are anticipated as a result of the community sessions. The potential activities include:

- Equip faculty fellows to work on their individual course projects using action research.
- Develop and administer computing proficiency evaluations to students, and carry out classroom observations (e.g., junior and senior project presentations) and portfolio (assignment) reviews, to evaluate the high-order skills we are targeting.
- Develop innovative computational assignments, projects, etc., as well as grading rubrics that faculty will use in their instruction to teach computing concepts.
- Form faculty fellow-industry partner pairs in which each pair co-develops a relevant assignment or project that emphasizes the computing skills essential to their respective industry.
- Develop assignments that emphasize a data-centric approach to engineering education; students needing to collect, process, analyze and communicate data is a common experience across engineering disciplines and computing plays a vital role in this activity.
- Create, gather and distribute innovative computing assignment ideas and materials, similar in spirit to the ones presented in the annual ACM-SIGCSE conference and collected by the Nifty Assignments project.[9] Following the guidelines in [9], nifty assignments will be designed to fit into the curriculum, be easy to adopt by other instructors, and incorporate elements of fun that invite students to "play around" with the material.
- Explore the possibility of defining a set of "computing fluency" standards relevant to high-level engineering problem solving (as opposed to lower level "computing literacy" standards).

The projects, assignments and associated materials will be distributed via a Web forum to other faculty at this university and elsewhere. Creating great assignments is a challenging and time-consuming part of course development. Sharing assignments and their materials will encourage and inspire other faculty to incorporate computing assignments in their disciplines. More information about the community pilot semester will be available for inclusion in the conference presentation.

3.2 Steady State

In keeping with the structure of the aforementioned Campus Writing and Speaking Program, the community coordinator will invite up to 12 new faculty members per semester from a range of engineering disciplines to participate in the community during the second year of the project. Participants will commit to attending two 2-hour sessions per month and to integrate computing into one of their courses by developing an assignment, lecture, in-class exercise, or etc. An industry panel, comprised of a new set of 4-6 representatives, will be formed each semester and will be asked to attend at least one session per month. The graduate student assistant trained during the planning phase will be available to provide personal attention to faculty participants to aid in executing the ideas generated and for assistance in developing relevant assessment measures. The semester will culminate with a poster session and/or some form of dissemination of information to other participants as to what was achieved in their respective disciplines. Administrators, faculty and computing staff will be invited to the poster session. With the administrative support of one of the Vice-Provosts at the university, a formal recognition program for the faculty fellows will be initiated. Each faculty participant will receive a stipend upon successful completion of a semester-long session and will graduate to the "faculty fellow" status. This status designation will encourage their continued participation in the community in subsequent semesters in an effort to have previous participants share their experiences with new participants. Disseminating information to those outside of the community is an integral part of this project. Faculty fellows will serve as the primary vehicles for circulating information back to the departments.

3.3 Preparation for Transformation and Broadening the Community

Continuing from Year 2, new faculty members will be invited to participate in the community, new industry panels will be formed, and the semesters will progress in a "business as usual" fashion in Year 3. At this point in the project, we anticipate broader and more systemic projects to be undertaken by the community, such as:

- Devise new computing-intensive courses across the College co-taught by two or more engineering departments; these courses will be centered on increasing the relevance of the undergraduate computing education to professionals in the field.
- Plan a multidisciplinary capstone senior design project course for all departments in the College of Engineering to engage computing intensive disciplines with other engineering disciplines.
- Create a junior year multidisciplinary project that relies on extensive collaboration with computing intensive disciplines.
- Initiate formal computing concentrations in select engineering departments.

In addition, a primary thrust of Year 3 will be on summarizing the results of the first two years of the project and preparing for an undergraduate computing education transformation plan. Efforts of the community that are successfully implemented at this university will be integrated into models that can be replicated at other institutions. Faculty fellows will convene at retreats to create these models and to discuss the potential for the community to bring about wide scale change to the system, not just modifications to existing curricula or courses at one or two universities. Year 3 will also be a year of outreach as we seek to broaden the community. Faculty fellows will attend conferences to make presentations to their peers on the results of the community building efforts at this university. The community coordinator plans to organize a national symposium where faculty fellows will present their strategies to transform the undergraduate computing education to other nationally recognized computing leaders. The findings will be submitted for publication as well as disseminated to the computing community at large.

Sustainability Efforts: Our goal is to create a lively and engaging community of stakeholders that remains active well beyond the initial three-year NSF funding period; indeed, it is unlikely that most of the issues facing computing education today will disappear in the next few years. We also recognize that one of the most effective methods for sustaining faculty involvement is to ensure the availability of resources for incentives and training opportunities; also, the community is unlikely to be sustained without recurring funds to cover operational expenses, graduate student time, publicity efforts, etc. We will explore several avenues for ensuring the availability of funding for our community beyond the third year. Supplemental funds can be made available through the College of Engineering's Student-Owned Computing Program orchestrated by the Office of the Dean to ensure future support of the community. We will also seek continued funding from the Provost and corporate sponsors. One specific approach we plan to pursue is to submit a proposal to our industry partners for the establishment of endowments to fund corporate-named fellows.

4. Assessment Plan

To ensure that the objectives of the project are met successfully, we are incorporating an assessment process as an integral part of our activities. There will be self-evaluation, as well as formative and summative evaluation of progress, at various times during the project period. In the remainder of this section, we outline a number of stages that will be included in the assessment plan as drafted by the director of assessment.

As part of our assessment plan, in the first semester of the project we began by conducting a baseline survey of faculty attitude toward, and use of, technology among the community participants and over 60 other faculty members from across the College of Engineering. The data obtained will be compared to a second survey implemented at the end of the project.

As faculty work in the community, they will be requested to complete journal entries during the semester to monitor what they are doing, what they are learning, and what impact it is having on their teaching. Minutes of all community session will be kept by the community coordinator and posted to the project website as evidence of issues encountered and progress made. This information will help with ongoing self-evaluation.

The director of assessment will conduct a formative evaluation at the end of Year 1. This will essentially be an implementation evaluation and will consider issues such as: (a) how many faculty are actively engaged in the community, (b) the degree of industry participation, (c) the extent to which community goals established at the beginning of the year were achieved, and (d) the benefits accrued from project implementation (personal, instructional, curricular, and systemic). Community participants and industry representatives will be surveyed and/or interviewed as part of this process, beginning with the survey the industry workshop participants completed.

Students will also be surveyed at the beginning and end of a semester to establish their confidence levels with various computing application processes relevant to particular courses. (A survey to this effect has already been developed and used within one engineering department, and can be modified for other applications, including for the purposes of this project). In classes of community participants, direct evidence of student performance on computing knowledge and skills will be obtained where possible.

In this formative process, we will pay special attention to the impact of the project. This early impact evaluation will consider issues such as: (a) what new curriculum documents have been put in place, (b) what research outputs (e.g., faculty poster presentations) have been generated, and (c) what the perceptions of the project are by administrators and other faculty. This information will help to inform participants in subsequent years.

The evaluation process in Year 1 will culminate with a workshop run by the director of assessment with the inaugural faculty fellows, where the findings will be discussed and suggestions for improvement and further self-evaluation in Year 2 developed.

In the second year of the project, we will implement an evaluation process similar to that in Year 1. In addition, at the end of the year, the original faculty fellows will hold a retreat with the faculty participating in the community activities during the year, as well as key administrators. The purpose of the retreat will be to discuss the two cycles of evaluation data, and to develop further strategies for increasing the impact of the project across the College of Engineering.

The self-evaluation processes used in Years 1 and 2 will continue in Year 3, as will the formative processes used for assessing the community.

The main thrust of the evaluation at the end of Year 3 will be summative in nature, focusing on the impact that the project has had. Factors considered in this evaluation will include: (a) the quantity and quality of faculty engagement in the community, (b) the quantity and quality of industry participation, (c) the nature and extent of benefits accrued to the participants (including industry participants), (d) the number and quality of research posters, presentations and publications made by faculty, (e) the nature and type of teaching and curriculum changes (e.g., new syllabuses; common outcomes developed for courses taught by different instructors; new innovative teaching activities and materials developed), (f) the degree to which technology and computing have been integrated across programs from freshmen to senior levels, and (g) the extent of support by administrators.

Data from student surveys and direct evidence of student performance over the 3 years will be analyzed and summarized. Questions will also be added to the Graduating Senior Survey to track student perceptions about: (a) computer fluency, (b) technology integration into engineering courses, (c) extent to which technology was used across their degree program, and (d) the role of technology and computing in enhancing aspects of learning, such as problem solving and critical thinking. This data will be collected annually to monitor changes in student perceptions from before the project is implemented to after 3 years of project implementation. At the end of the project, the director of assessment will compile a summative evaluation report.

5. Concluding Remarks

This project brings together a highly qualified group of faculty and staff with a broad range of experiences in academic computing, instruction and research. The team will partner with local industry leaders and together will provide an insightful vision into the future of computing fluency in the engineering workplace and how it should be nurtured on the university campus. It is our hope that effective strategies will be developed based both on local needs and the existing literature on academic computing. It is our intent for the results of this project to provide a vehicle for systemic institutional change in academic computing in engineering.

Bibliography

1. Campus Writing and Speaking Program. North Carolina State University. 21 February 2008 <http://www2.chass.ncsu.edu/CWSP/>.
2. Wing, J. "Computational Thinking." *Communications of the ACM,* 49(3) 2006: pg. 33-35.
3. Reilly, C. "Teaching by example: A case for peer workshops about pedagogy and technology." *Innovate*, 1(3) 2003.
4. Linstone, H.A. & Turoff, M. The Delphi Method: Techniques and Applications. 1975. 4 January 2008 <http://is.njit.edu/pubs/delphibook/delphibook.pdf>.
5. McKitrick, S. "Assessing 'ineffable' general education outcomes using the Delphi approach." North Carolina State University Assessment Symposium: Cary, NC 2007.
6. Tague, N. "Affinity Diagram." 2004. 20 February 2008 < http://www.asq.org/learn-about-quality/idea-creation-tools/overview/affinity.html>.
7. Hopkins, D. A Teacher's Guide to Classroom Research. Philadelphia: Open University Press, 1985.
8. Brent, R. & Felder R.M. "Engineering faculty development: Getting the sermon beyond the choir." *Journal of Faculty Development*, 18(3) 2001: pg. 73-81.
9. Parlante, N. Nifty assignments. Stanford University. 10 January 2007 <http://nifty.stanford.edu/>.