CPATH-CB: Computing Across Curricula

Project Summary

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 in the 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. Rather, 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 computing in the workplace, we must begin to think in terms of pervasive computing in their education.

The overarching goals of this project are to: (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. To this end, this project will establish 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 will be modeled in structure after the nationally renowned NC State Campus Writing and Speaking Program (CWSP), and will leverage our experience with the NSF-funded SUCCEED coalition, driven by and headquartered at NC State, as well as NC State's Student-Owned Computing initiative.

<u>Intellectual Merit</u>

The project brings together a highly qualified group of faculty and staff with a broad range of experiences in academic computing, instruction and research. This 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. Effective strategies will be developed both based on local needs and the existing literature on academic computing. The work will also be informed by the latest findings in the learning sciences as promoted by NSF and other national leaders in STEM education.

<u>Broader Impact</u>

The results of this project will provide a vehicle for systemic institutional change in academic computing in engineering. Since project leadership includes key members of existing academic computing units on campus, the project is set to implement these reforms from inside of existing institutional structures. In addition, a primary thrust of this project is to establish a standing group on campus where faculty and industry leaders can come together and continue to effect change in academic computing after initial project funding has concluded. Results of the project will be communicated both locally on campus through the above-mentioned vehicles, but also nationally through regional and national meeting and journals. Project leaders are active members in national STEM education organizations such as ACM-SIGCSE and ASEE.

1 Vision, Goals, Objectives, and Outcomes

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 workplace. Computing in the workplace, and of particular interest in our case the engineering workplace, is pervasive and 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 graduate 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 our students for pervasive computing in the workplace, we must begin to think in terms of pervasive computing in the education of our students. The overarching goals of the proposed activities are to:

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.

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.

To this end, we will establish 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. Our "Computing Across Curricula" (CAC) community will be modeled in structure after the nationally renowned NC State Campus Writing and Speaking Program (CWSP), and will leverage our experience with the NSF-funded SUCCEED coalition, driven by and headquartered at NC State, as well as NC State's Student-Owned Computing (SOC) initiative (for information on SUCCEED and SOC, please refer to Section 3). The Campus Writing and Speaking Program was established in July 1997 to provide leadership for NC State as it implements its General Education Requirements which state that "each curriculum is designed so that upper-level courses and other programmatic experiences help students write and speak competently in the disciplines." During this time, the CWSP has successfully implemented a wide range of writing and speaking strategies across the curriculum. One of the program's specific goals is "to provide faculty with resources that help prepare students to become proficient writers and speakers in their respective disciplines" [3].

Whereas the CWSP focuses on communication skills, a shift in focus to computing skills allows for logical translation of this program to the community proposed here.

1.1 The Computational Thinking Thread

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 called 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 the 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 students 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.

1.2 The Computing Across Curricula (CAC) Community

Faculty learning communities are nothing new to academe. The goal of the efforts outlined herein is to build off of a proven faculty collaboration method. To date, many learning communities have been limited to faculty involvement only, and involvement from engineering faculty in these formal university-wide communities has been somewhat disappointing. In contrast, industry outreach and involvement are a critical component of the proposed community building project. 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 meeting our goals. Indeed, data from the NC State alumni survey conducted in spring 2006 indicate 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 so as to open up meaningful channels for dialogue and for ideas to flow from industry to the university. We have shared this vision with potential industry partners and their reaction has been enthusiastic; letters of support from several prominent companies are included with this proposal.

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 team of PIs 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 [18] have shown that the existence of even a small number of faculty within each department with expertise on integrating technology and pedagogy is sufficient to drive wide-spread adoption of similar practices across a university (refer also to the relevant discussion in Section 3). Therefore, we expect that faculty fellows will 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, however, that this three-year 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 proposal, but will be considered as a follow-up activity.

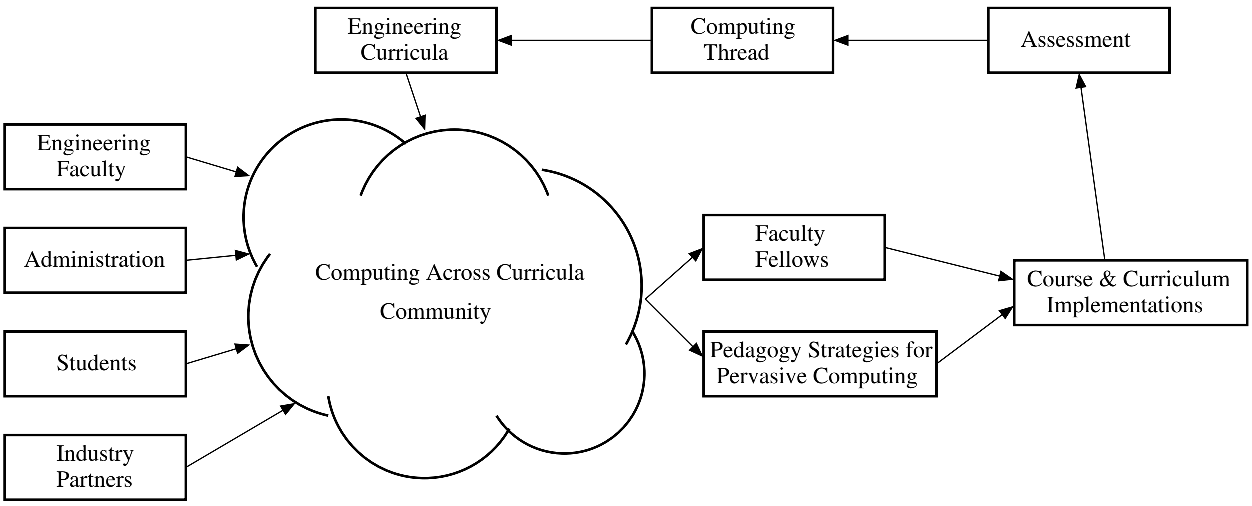

Figure 1: Computing Across Curricula (CAC) Community

1.3 Objectives and Outcomes

Specific objectives of the proposed community building 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 styles 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 faculty activities 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.*
  During the third year of the project, the community will undertake broader and 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. The preparation of a CPATH Transformation proposal is a possible outcome of these activities.

- *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.

In the following section, we outline the activities we will undertake to realize the goals and objectives of the project, as well as an associated assessment plan. In Section 3, we provide a survey of related activities nationally and at NC State, and we discuss results from prior work at NC State in the context of related NSF projects.

2 Implementation and Assessment Plans

The community activities and implementation plan we describe in this section will be carried out by an interdisciplinary team of PIs and Senior Personnel affiliated with the Colleges of Engineering and Education, including the College of Engineering Director of Assessment (Dr. Raubenheimer) and the Coordinator of SOC (Craig). The faculty participating in this proposal (Bullard, *Chemical and Biomolecular Engineering*, Joines, *Textile Engineering and SOC chair*, T. Miller, *Electrical and Computer Engineering and Vice Provost for Distance Education and Learning Technologies*, C. Miller and Rouskas, *Computer Science*, Silverberg, *Mechanical and Aerospace Engineering*, and Wiebe, *Math, Science, and Technology Education*) collectively have a wide range of experience in both discipline-specific learning technologies and broadly defined computing education. In addition to the complementary strengths and points of view that each PI and Senior Personnel brings to the table, various pairings or subsets of the team have a proven record of collaboration in terms of integrating technology and teaching, carrying out related assessment plans, and disseminating the results within and outside NC State. By drawing upon the success of, and our collective experience with, SUCCEED, SOC initiative and forums, and the Campus Writing and Speaking Program (CWSP), we plan to build an engaging community and sustain it well beyond the three-year funding period. Hence, we believe that we are in a unique position to carry out the proposed work and have an impact on revitalizing undergraduate computing education.

Our activities will initially target faculty from across the College of Engineering and other CISE disciplines. 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 proposed CAC community campus-wide, similar to the CWSP program.

2.1 Year 1: Community Formation, Organization and Management

Our focus during the first year of the project will be on relevant planning and pilot activities that will set the stage for the various community events and functions in following years. The

Coordinator of SOC (Craig) will serve as the inaugural community coordinator, and will organize all gatherings and oversee the various activities. The faculty PIs and Senior Personnel will form the base of the multidisciplinary community, and will serve as inaugural faculty fellows.

Planning. During the first semester of Year 1, the community director and faculty fellows will engage in the following activities:

- formalize the community vision and mission by developing a community charter;
- develop a plan of action using this proposal as a reference;
- align the plan of action with the formal evaluation plan, in coordination with College of Engineering Director of Assessment;
- contact other industry partners to solicit their participation;
- generate potential projects to be undertaken by the community; and
- train the graduate student assistant(s) who will be involved in the community activities.

Community Pilot. In the second semester, 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 community coordinator. The industry panel will be initiated during the pilot semester, and 4-6 industry representatives will be included in the panel. Industry involvement is a critical component to 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.

Session formats and topics will vary. 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 members 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 will be an integral part of the community. To ensure community sessions are an appropriate mix of structure and flexibility and remain broad in scope, the community coordinator will organize and facilitate all community sessions and will employ a variety of research and discussion methods. Brainstorming and nominal group technique (NGT) [10] will be employed where appropriate to initiate discussions. Faculty will use the iterative process associated with action research to test out potential strategies. According to Kemmis, as cited in [15]: *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.* Furthermore, the planning of the various community meetings, seminars, and workshops will be guided by the following three principles that were discovered during the SUCCEED project and have been reinforced by several other studies [7]:

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.

A broad range of projects and initiatives are anticipated as a result of the community meetings:

- 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 representative 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 key role in this activity.

- Create, gather, and distribute innovative computing assignment ideas and material, similar in spirit to the ones presented in the annual ACM-SIGCSE conference and collected by the Nifty Assignments project [4]. Following the guidelines in [4], 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 material will be distributed via a Web forum to other faculty at NC State 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.

2.2 Year 2: Steady State

In keeping with the structure of the CWSP, the community coordinator will invite up to 12 new faculty members (1 per discipline) per semester to participate in the community. Participants will commit to attending two 2-hour sessions per month facilitated by the community coordinator, and to utilizing computing into one of their courses by developing an assignment, lecture, or in-class exercise. A new industry panel, comprised of a new set of 4-6 representatives from industry, will be formed each semester and will be asked to attend at least one session per month (approximately four sessions per semester). Through support of the NC State Industrial Extension Service (IES), volunteers from local chapters of several professional organizations will also be invited to participate in community activities.

The graduate student assistants trained during the planning phase will be available to provide personal attention to faculty participants and to aid in executing the ideas generated. 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 discipline. Administrators, faculty, and computing staff will be invited to the poster session.

With the administrative support of the Vice-Provost for Distance Education and Learning Technology, a formal recognition program for faculty participants will be initiated. Faculty participants will receive a stipend upon successful completion of a semester-long session and will graduate to the "faculty fellow" status; faculty stipends are included in the submitted budget as "contracted services." 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 the community. Faculty fellows will serve as the primary vehicles for flowing information back to the departments.

2.3 Year 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" state in Year 3. At this point in the project, we anticipate much broader and systemic projects to be undertaken by the community:

- 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 courses for all departments in the COE

to engage CISE disciplines (particularly Computer Science) with other engineering disciplines.

- Create a junior year multidisciplinary project that relies on extensive collaboration with CISE 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; it is our hope and ultimate objective that this plan will become the basis for a CPATH Transformation proposal to the NSF at that time. Efforts of the community that are successfully implemented at NC State will be integrated into models that can be replicated at other institutions. Faculty fellows will again 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 NC State. The community coordinator will also 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 three-year 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 Student-Owned Computing program, orchestrated by the College of Engineering 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 corporate partners for the establishment of endowments to fund corporate-named fellows.

2.4 Assessment Plan

To ensure that the objectives of the proposed project are met successfully, we plan to incorporate 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. We emphasize, however, that this plan is preliminary. If the project is funded, the Director of Assessment in the College of Engineering (Raubenheimer) will develop a detailed assessment plan together with the project PIs, and will conduct the formative and summative evaluation processes.

Year 1. At the beginning of Year 1, we will conduct a baseline survey of faculty attitude toward, and use of, technology among the community participants, and of as many other faculty as possible within the College of Engineering. This data 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 meetings will be kept by the community coordinator 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.

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 the Department of Chemical and Biological Engineering, 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 include a workshop run by the Director of Assessment with the PIs, where the findings will be discussed and suggestions for improvement and further self-evaluation in Year 2 developed.

Year 2. In Year 2, we will implement an evaluation process similar to that in Year 1. In addition, at the end of Year 2, the PIs 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.

Year 3. 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 that will be included in the final project report to the NSF.

Table 1 summarizes the planned activities and provides a tentative project timeline.

3 National Discussion and Results from Prior NSF Support

In the fall of 1996, the Higher Education Group at Microsoft held a meeting of its Scholars board to ponder the question *"What are the major factors which inhibit the accelerated adoption of technology in higher education?"* [17]. At the end of the meeting, they agreed on 18 different factors that have been proven to be germane across universities in inhibiting the use of technology in the classroom. Despite all the progress since then, ten years later universities still face many of the same 18 factors documented in [17]. Most progress so far has been with tackling factors related to the technology itself or the infrastructure needed to initiate, maintain and update the technology. What has lagged tremendously is the degree to which faculty integrate new technology with instruction [17]. As a result, new technology is often thrust upon faculty, without a clear understanding of how it can complement or enhance the educational process. At NC State, there have been a number of initiatives aimed at assisting faculty in integrating computing into their courses, including the learning in a technology-rich environment (LITRE) program [2] and the student-owned computing (SOC) initiative [5] within the College of Engineering; SOC completed a five-year laptop pilot program exploring the benefits of mobile computing on academic instruction.

As a whole, faculty utilize computers and technology extensively on a daily basis to send e-mail, use word processing programs, search the Web, or in their own research. However, the number of faculty members using technology to enhance their teaching is relatively low. Green [11] reports that the Campus Computing Project stated that over 600 two- and four-year public and private universities in the United States "identify 'assisting faculty integrate technology into instruction' as the single most important IT issue confronting their campuses 'over the next two or three years'." However, two years later, Green [12] reported from the same survey that only a third of faculty use course management tools for online course resources, while just a third actually place materials on the Web despite wide availability of technology and support services. More recently, the *EDUCAUSE Core Data Service Fiscal Year 2004 Summary Report*, a survey about campus information technology environments at 890 colleges and universities in the U.S. and abroad, found that "only 19% of all campuses report that these systems are for all or nearly all courses" [14].

Table 1: Project Timeline

Activity/ Responsible party†	**Year 1**		**Year 2**		**Year 3**	
	Sem. 1	**Sem. 2**	**Sem. 3**	**Sem. 4**	**Sem. 5**	**Sem. 6**
Planning						
Initial community formed	■					
Community charter	■					
Plans for pilot structure	■					
Initial project strategies	■					
Formal assessment plan	■	■				
Group: Initial FF, CC, DA, AM						
Community Pilot						
Industry outreach	■	■				
Pilot sessions held		■				
Publicity efforts	■	■	■	■	■	■
Community plans refined		■				
Group: FF, CC, DA, IP						
Community "Steady State"						
Regular sessions held			■	■	■	■
Information dissemination		■	■	■	■	■
Group: FF, CC, NF, IP						
Preparation for Transformation						
Successful projects documented				■	■	■
Integrative new models formed					■	■
CPATH transformation proposal						■
Group: AM, FF, CC, NF, IP						
Community Broadening						
Findings submitted for publication					■	■
Other Universities participate					■	■
Host national symposium						■
Group: AM, FF, CC, NF, IP						
Assessment Activities						
Assessment plan development	■	■				
Faculty surveys and self-evaluation		■	■	■	■	■
Student surveys		■	■	■	■	■
Assessment workshop/retreat		■		■		■
Year 1 assessment review and report		■				
Year 2 assessment review and report				■		
Year 3 assessment review and report						■
Group: DA, FF, CC, NF						

†AM: Administration, CC: Community Coordinator, DA: Director of Assessment, FF: Faculty Fellows, IP: Industry Partners, NF: New Faculty Participants

If the technical infrastructure, support staff, as well as some training opportunities exists then what is impeding faculty from embracing technology to enhance their courses? Gustafson [13] states a lack of time, a lack of skills, and a general discomfort with using computing technology in a teaching environment (i.e., pedagogical issues) as the most often reasons given by faculty for not using it.

Kopyc [16] states that while several universities are trying to involve faculty and IT together to develop workshops for faculty, or convening committees to try foster faculty engagement with technology, there needs to be other strategies that attack the fundamental issue of pedagogy and adoption. One strategy undertaken by Bard College is the development of Teaching Forums consisting of faculty, computing staff, and administration. However, the faculty led the discussions and directions to eliminate the tendency of staff driving the classroom technology. Through the forums, faculty have the opportunity to share their various concerns about technology and its direct impact on teaching pedagogy and student learning while providing feedback to the administration and staff. Several faculty-led workshops were generated from these open forums.

At NC State, the initial objective of the Student-Owned Computing (SOC) task force in the College of Engineering was to launch a student-owned platform and support plan [5]. After accomplishing its original goal, in 2003 the group changed direction to pursue a new objective: "to explore how laptop computers and wireless connectivity can enhance undergraduate engineering education." To this end, there have been many initiatives to engage professors and instructors, both within and beyond the College of Engineering, to embrace technology in teaching. Several of the PIs and Senior Personnel (Bullard, Joines, C. Miller, Rouskas) were among the first to participate in these initiatives, and have redesigned sophomore- and junior-level courses in their respective departments by adapting learning materials to: use the available technology, provide a pedagogy utilizing more learning styles than traditional lectures, and provide an interactive learning experience for each student. As Chair of SOC, one of the PIs (Joines) has organized ten SOC forums over the past two years to disseminate the use of computer technology to enhance the undergraduate teaching and learning. These forums were intended to give discipline-specific examples of good practices, be a medium for exchange of ideas, as well as disseminate information about tools that can be used to enhance student learning. The main challenge with these volunteer forums has been to attract interest beyond a core group of "believers": faculty attendance has varied significantly from one meeting to another, with many of the same individuals attending most forums.

A similar initiative at Lehigh University is the Lehigh Lab whose mission is to "provide a locus for faculty and students to advance the adoption of innovative technologies and techniques that enhance teaching, learning, and research" [1]. According to their web page, "the Lab concept is founded upon the idea that the University as a whole is a laboratory in which faculty, staff and students work and experiment together, across departments and disciplines, to advance learning." One important aspect of the Lab's strategy is the support of a technology fellow program where faculty can apply for release time to develop skills and create technology-enhanced courses; such an approach provides recognition for involved faculty, rewards them for their efforts, and demonstrates administrative support. Consequently, it addresses one of the greatest obstacles, as cited by surveys of faculty members at large public universities [9], to using or experimenting with technologies for enhancing student learning: lack of time.

Having even a few faculty members within each department develop expertise in using technology to enhance teaching may have important and material impact on wide-spread adoption across a university. These technology liaisons understand the disciplinary norms, can speak with their colleagues about concrete examples on how and which technologies may benefit them, as well as know the best and most appropriate applications within their own specific discipline. These liaisons/peers represent a resource for other faculty within the department in addition to the traditional computing staff, especially on issues of pedagogy [18].

Computer literacy standards in the U.S. are only loosely defined. The results of a Google search on the subject indicate that many of the community college systems across the country are well ahead of universities in defining computer competency standards for their students. Most require students to pass a test or take certain classes to demonstrate a certain level of understanding. This state of affairs is likely due to the fact that the main mission of community colleges is to train students directly for certain jobs. Also, many states have computer competency standards for their secondary school students before graduation. Computer competency, as defined by these constituent groups, usually refers to three general areas: (1) computing fundamentals (e.g., hardware, software, using the operating system), (2) key applications (e.g., word processing, spreadsheet functions, presentation software, using software), and (3) navigating the internet (e.g., e-mail, searching, instant messaging, etc).

In the College of Engineering at NC State, the vast majority of our students meet these low-level computer competency standards. Furthermore, all incoming College of Engineering freshmen are required to complete the introductory computing course E115. The purpose of this course is to ensure the same level of computer competency among all students. The course was revamped in the Fall of 2005 to reflect the new expectation that "all incoming undergraduate students have a laptop computer that meets college requirements" [5]. The students now learn to maintain and operate their own laptops and to complete assignments using NC State's course management systems, in addition to the skills mentioned above. With this project, we aim to take the first steps towards developing students that possess a higher level of "fluency," i.e., the ability to think critically about modeling, analyzing, and solving problems by drawing from a wide palette of sophisticated computing tools.

SUCCEED. In 1991, the National Science Foundation began funding coalitions of engineering schools to develop, implement, institutionalize, and disseminate reforms in engineering education. In the second year of the program, the NSF funded SUCCEED, a coalition of eight institutions in the Southeastern United States (Clemson, Florida A&M/Florida State University, Georgia Tech, North Carolina State, North Carolina A&T, University of Florida, University of North Carolina at Charlotte, and Virginia Tech). SUCCEED was driven by and headquartered at NC State. In its first five-year funding period, SUCCEED focused primarily on developing innovative teaching materials and programs, including integrated first-year engineering curricula, instructional modules and delivery tools for technology-based courses, programs to promote writing and design across the curriculum, and programs to promote recruitment and retention of minorities and women.

In 1997, SUCCEED was awarded funding for an additional five years, with its mission shifting from program development to scale-up, institutionalization, and dissemination of the innovations developed in the first funding period. Moving the innovations developed in the first five years into the mainstream curriculum required the involvement of many more faculty members, so in 1997 SUCCEED initiated a coalition-wide faculty development (FD) program. The FD program goals were to design and implement a model for sustainable faculty development in engineering on all Coalition campuses, and to involve at least 60% of the Coalition faculty in FD offerings by the end of the second five-year period. At the end of the tenth year of funding, the faculty development model had been formulated, implementation was well under way at all Coalition campuses, and faculty participation in FD programs had exceeded its target level [6–8]. NC State has continued with many of the FD programs originated out of SUCCEED, demonstrating that the university is committed to sustaining effective programs. One of the FD programs that not only continues today but has expanded to many colleges beyond the College of Engineering, is the new faculty orientation workshop. This five-day workshop covers a wide range of topics including the development of faculty research programs, teaching pedagogies and effectiveness, and other aspects of academic life that new faculty must navigate successfully in order to succeed in a demanding academic environment.

References

[1] Lehigh university Lehigh lab. http://www.lehigh.edu/lehighlab/.

[2] LITRE: Learning in a technology-rich environment. http://litre.ncsu.edu/.

[3] NCSU campus writing and speaking program. http://www2.chass.ncsu.edu/CWSP/.

[4] Nifty assignments project. http://nifty.stanford.edu/.

[5] Student owned computing. http://www.eos.ncsu.edu/soc/.

[6] C.E. Brawner, R.M. Felder, R. Allen, and R. Brent. 1999-2000 succeed faculty survey of teaching practices and perceptions on institutional attitudes toward teaching. http://www.ncsu.edu/felder-public/Papers/99faculty_survey.pdf, 2001.

[7] R. Brent and R.M. Felder. Engineering faculty development: Getting the sermon beyond the choir. *Journal of Faculty Development*, 18(3):73–81, 2001.

[8] R. Brent and R.M. Felder. A model for engineering faculty development. *Journal of Faculty Development*, 19(2):234–240, 2003.

[9] J.F. Chizmar and D.B. Williams. What do faculty want? *Educause Quarterly*, 1:18–24, 2001.

[10] A. L. Delbecq and A. H VandeVen. A group process model for problem identification and program planning. *Journal Of Applied Behavioral Science*, (VII), July/August 1971.

[11] K.C. Green. The 2001 national survey of information technology in u.s. higher education: eCommerce comes slowly to campus. http://www.campuscomputing.net/summaries/2001/index.html, 2001.

[12] K.C. Green. The 2003 national survey of information technology in u.s. higher education: Campus policies address digital content and copyright; wireless networks show big gains. http://www.campuscomputing.net/summaries/2003/index.html, 2003.

[13] K. Gustafson. The impact of technologies on learning. *Planning for Higher Education*, 32(1):37–43, 2003.

[14] B.L. Hawkins, J.A. Rudy, and R. Nikolich. Educause core data service fiscal year 2004 summary report. http://www.educause.edu/ir/library/pdf/pub8002b.pdf, 2004.

[15] D. Hopkins. *A Teacher's Guide to Classroom Research*. Open University Press, Philadelphia, 1985.

[16] S. Kopyc. Enhancing teaching with technology: Are we there yet? *Innovate*, 3(2), 2006.

[17] J. G. Ptaszynski. Shared misery/shared solutions: Major factors inhibiting the accelerated adoption of technology in higher education. http://technologysource.org/article/shared_misery_shared_solutions/.

[18] C. Reilly. Teaching by example: A case for peer workshops about pedagogy and technology. *Innovate*, 1(3), 2003.