Aligning Computing Education with Engineering Workforce Computational Needs: New Curricular Directions to Improve Computational Thinking in Engineering Graduates

Claudia E. Vergara, Mark Urban-Lurain, Cindee Dresen, Tammy Coxen, Taryn MacFarlane, Kysha Frazier, Daina Briedis, Neeraj Buch, Abdol-Hossein Esfahanian, Louise Paquette, Jon Sticklen, Jeannine LaPrad, Thomas F. Wolff

vergara@msu.edu, urban@msu.edu, cdresen@skilledwork.org, tcoxen@skilledwork.org, tmacfarlane@skilledwork.org, kfrazier@skilledwork.org, briedis@egr.msu.edu, buch@egr.msu.edu, esfahanian@cse.msu.edu, paquetl@lcc.edu, sticklen@msu.edu, jmlaprad@skilledwork.org, wolff@egr.msu.edu

Abstract - **In this global economy, the preparation of a globally competitive U.S. workforce with knowledge and understanding of critical computing concepts is essential. Our CPACE (Collaborative Process to Align Computing Education with Engineering Workforce Needs) vision is to revitalize undergraduate computing education within the engineering and technology fields. Our objective is to design and implement a process to engage stakeholders from multiple sectors and identify the computational tools and problem-solving skills and define how these skills--directly informed by industry needs--can be integrated across disciplinary curricula. By explicitly integrating computing concepts and disciplinary problem solving, engineering graduates will enter the workforce with improved and practice-ready computational thinking that will enhance their problem-solving and design skills. We present the analysis of the computational skills and the strategies that we are using to map the workforce problem-solving requirements onto the foundational computer science principles. We outline the framework that we are using to identify opportunities for curricular integration between computer science concepts and the disciplinary engineering curricula. By documenting, evaluating, and making the process explicit, this process can serve as a model for national efforts to strengthen undergraduate computing education in engineering.**

Index Terms – CPACE; Computing education in engineering; Engineering curricular reform.

PROJECT OVERVIEW

Engineering schools are responsible for the preparation of a workforce with the analytical reasoning and practical skills that will allow them to deal with complex forms of professional practice in an ever-changing global environment. Engineering professionals recognize the need to move from an educational scheme that primarily emphasizes the acquisition of technical knowledge distantly followed by the contextual social professional practice to an approach that allows students to integrate conceptual knowledge, technical skills and professional practice [1].

Michigan State University (MSU), Lansing Community College (LCC), and the Corporation for Skilled Workforce (CSW) are collaborating to address the preparation of new engineers in one aspect of their professional practice: computational thinking. According to Jeannette Wing, "computational thinking" 'represents a universally applicable attitude and skill set' fundamental for everyone [2]. Broadly defined and in the context of the engineering practice, computational thinking involves solving problems and designing systems by drawing on fundamental computer science concepts rather than simply thinking about computing as a calculation tool. It is a vital component in the preparation of engineering students who are able to meet the challenges of this global economy.

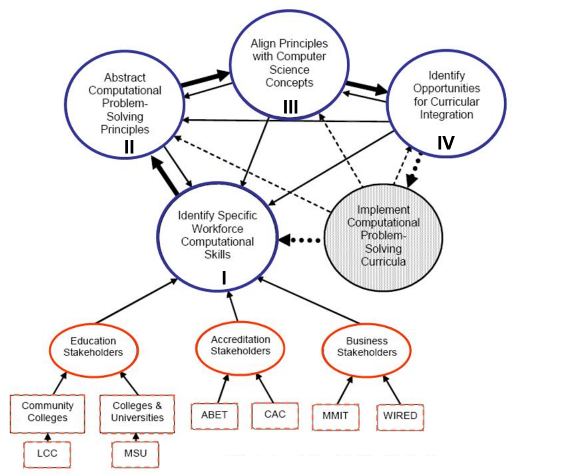
To address these needs, MSU, LCC and CSW partnered on a **C**ollaborative **P**rocess to **A**lign **C**omputing **E**ducation with Engineering Workforce Needs (CPACE). The project objective is to design and implement a process to transform undergraduate computing education within the engineering and technology fields. Our primary emphasis is to engage stakeholders from multiple sectors in the collaborative transformation of computing education within higher education institutions. The project engages computer science and engineering faculty from a community college and a research university with regional technology and engineering employers to develop a working dialog focused on redesigning the disciplinary engineering curricula around computational tools and problem-solving competence *that is aligned with industry needs*.

In this paper, we summarize the process developed to engage these stakeholders [3]. We present and discuss the analyses from our employer interviews and employee surveys aimed to determine the stakeholder's assessments of the computational skills needs in their business sectors. We also present the strategies that we are using to map the workforce computational skills requirements onto the foundational computer science principles and outline the framework that we are using to identify opportunities for integration in the disciplinary engineering curricula.

IMPLEMENTATION STRATEGY

The implementation plan derives from a dynamic model that allows all stakeholders to see the interrelationships among components (Figure 1).

FIGURE 1



THE CPACE TRANSFORMATION MODEL. THE VARIOUS STAKEHOLDERS GROUPS AND SUBGROUPS ARE HIGHLIGHTED IN RED. THE THICK ARROWS CONECT THE VARIOUS STEPS OF THE PROJECT (NODES). THE FOCUS OF THIS PROJECT IS ON NODES I - IV. THE SHADED NODE INDICATES THE CURRICULAR IMPLEMENTATION PROCESS THAT WILL BE ADDRESSED IN A SUBSEQUENT PROJECT.

The process comprises five stages:

1. Interview and survey stakeholders to identify specific workforce computational skills.
2. Abstract computational problem-solving principles from those skills.
3. Map the problem-solving requirements onto underlying computer science concepts that are the foundation of the computer science discipline. This alignment is checked among stakeholders to confirm that we have captured important skills.
4. Identify opportunities for curricular integration that fit between the computer science concepts and engineering curricula in other departments. The abstract concepts are aligned with disciplinary problem-solving that address workforce needs.
5. Implement computational problem-solving revisions in engineering curricula.

We have divided the outcomes of the project into three strategic phases summarized in Table I. Each phase can be aligned with the nodes in the transformation model depicted in Figure 1. The table lists the activities required in each phase.

TABLE I
PROJECT OUTCOMES

Phase I (node I)	Phase II (nodes II, III and IV)	Phase III
- Design a work plan to identify and engage engineering stakeholders from various sectors to identify common practices around computing education. - Engage engineering stakeholders. - Develop interview and survey instruments to determine stakeholder's assessments of the computational skills needs in their business sectors. - Conduct interviews and to survey their engineering and technical employees	- Identify key computational problem-solving skills in these business sectors. - Abstract the computing principles and concepts and align those principles with computer science concepts that are the foundation of the computer science discipline. - Identify opportunities for curricular integration. - Synthesize and summarize a concise process from engagement to the identification of opportunities for curricular integration.	- Design and hold forums and other events to communicate findings from the process and begin planning for extensive engagement. - Engage a wider set of stakeholders in the preparation and submission of a full implementation grant.

An external evaluator assesses the project model and process at all phases and prepares reports of each phase of the activity.

Detailed information about the strategies that we are using to develop the process, including engagement, characteristics of the sample and development of the employer interview and employee survey instruments, can be obtained in Vergara et al 2009 [3].

In the following sections we present results from the analysis of the employer interviews and employee surveys developed for identifying computational skills. We also present a "map" that aligns key computational skills requirements in the engineering workplace to fundamental computer science principles and discuss how these skills may be integrated across engineering departmental courses.

EMPLOYER INTERVIEWS

Our objective in the interviews was to understand employers' perspectives on their engineering staff's use of computer technology and the computational skills needed in their businesses. We wanted to understand the employers' perception of how higher education is preparing their employees to meet the computing challenges they face and what recommendations for improvements might be made.

We conducted 27 interviews with human resources executives and the heads of engineering representing a cross-section of engineering disciplines and different industry sectors (Table II).

TABLE II
EMPLOYERS DISCIPLINE AND INDUSTRY SECTOR REPRESENTATION

Disciplines	Industries
- Applied Engineering Sciences - Bio-Science Engineering - Chemical Engineering - Civil Engineering - Computer Engineering - Computer Science - Electrical Engineering - Mechanical Engineering	- Software publishers - General manufacturing - Bio-products manufacturing - Aerospace Management and technical consulting services - Architectural and engineering services - Motor vehicle manufacturing - Government - Power generation and supply - Software publishers - Electronic instrument manufacturing

Employer interviews analyses

The employer interviews were organized and analyzed using *Transana* software that supports the transcription and analysis of large collections of video and audio data [www.transana.org]. We transcribed each interview and then coded responses by theme in each of three categories: general skills, computational skills, and future of engineering practice. These themes are summarized in Table III.

TABLE III
EMPLOYER INTERVIEW ANALYSES; COMMON THEMES

General Skills	Computational Aspects	Future of Engineering Practice
- Communication skills - Team work - Critical thinking - Innovative thinking - Problem solving (both conceptual and operational) - Ability to learn/adapt	- Basic computational skills. - Understanding of principles, application and limitations of computational tools - Using technology to collaborate at all levels - Use of technology to support broad problem solving and decision making - Familiarity with multiple software systems - Ability to move between abstractions in software and physical systems - Multiple CAD programs including 3D modeling - Process simulation packages - Numeric computational platforms - Excel (High level capabilities) - MS Office - Some programming	-Corporate development, leadership, management skills. - Project management software - Increasing integration of engineering data across larger systems - More business intelligence embedded in systems - Data Mining - Globalization - Environmental impact across disciplines. Design for the environment (DFE) - Research and development including: ▪ Material development/new applications for existing materials ▪ Electronic communication ▪ Next generation of technology - Increasing use of simulation to reduce materials usage in design phase

The analyses indicate that, in general, the employers place a high value on a) interpersonal skills (e.g., communication skills, the ability to organize and present data, and the ability to function in a team); b) critical and innovative thinking and problem solving; c) understanding business practices and the importance of integrating engineering data across larger systems to support globalization; and d) using computational tools to solve engineering problems by moving between abstractions in software and physical systems.

EMPLOYEE SURVEYS

To gain a complete picture of engineering computational needs, we also surveyed engineers and engineering technicians. Our main objectives were to 1) understand employees' perspectives on their uses of computing; 2) understand what they see as the strengths and weaknesses of their undergraduate computing education; and 3) identify current and future computational problem-solving gaps based on employee's views of future needs and trends. Our goal was to conduct electronic surveys of 250 employees of participating companies as well as leaders and members of the professional engineering societies. At the time of this writing, we have collected about 180 surveys.

The survey included the following sections:

- Demographics
- Skills and education: Questions about degrees and basic skill set needed for their job.
- Technology: Probes about the software that the company uses to meet key engineering challenges. We asked them to identify and describe up to three specific software programs they consider critical to support the engineering practices of their business and to the performance of their job.
- Future software directions: We requested specific examples about software that has changed the nature of their work and software functions that they anticipate will be critically important to support daily engineering practices over the next three to five years.

The majority of the questions in the survey were closed items, but it also contained some open-ended items to obtain detailed information about the employees' use of computing in the engineering practice.

Employee survey analyses

In this section, we present the analysis of questions that deal with the computational aspects of the survey; specific software programs that are considered critical to the engineering practice (mission-critical software). Our data include information about the key computational skills needed to use the mission-critical software. The open-ended responses were analyzed using the SPSS Text Analysis for Surveys [4].

Preliminary data indicate an agreement with the employer interviews. For example, we asked about the importance of a new engineer's knowledge of and skills in several specific tools or general categories of software. The responses are summarized in Figure 2; the general categories include specific software tools as follows:

1) Excel

2) Calculation platforms (MATLAB, MATHCAD, Mathematica, Maple)

3) CAD software

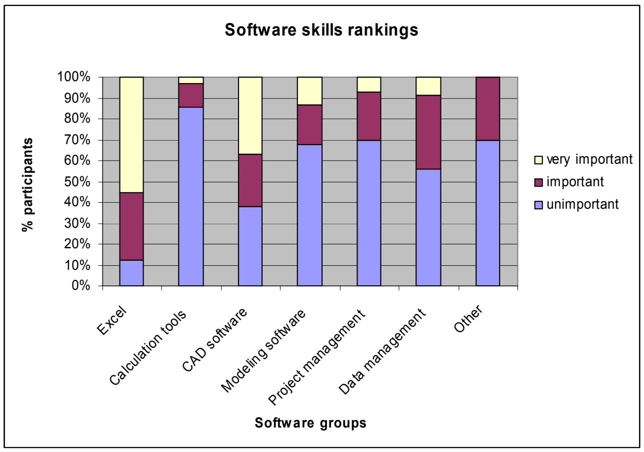
4) Modeling (3D modeling, finite element analysis, simulation packages)

5) Project management (requirements analysis and system architecture software, project lifecycle management (PLM) or product data management (PDM) software)

6) Data management (real time data collection/analysis software, statistical analysis software, graphics or imaging software)

7) Other (e.g., Micro-station, Project wise, Visual Studio, Geopack).

FIGURE 2



SOFTWARE SKILLS RANKING. ENGINEERS RANKING OF THE IMPORTANCE OF SPECIFIC COMPUTATIONAL TOOLS

We asked the employees about mission-critical software. The responses are summarized in Table IV.

TABLE IV
MISSION-CRITICAL SOFTWARE IN THE ENGINEERING PRACTICE

Software categories	% Total responses
CAD	34
Spreadsheet	19
Industry specific	18
Office/productivity	9
Project management	5
Programming	3
database	3
PLM (Product Lifecycle Management)	3
Mathematical software	2
Process Engineering	2
Statistical	1
Proprietary	1

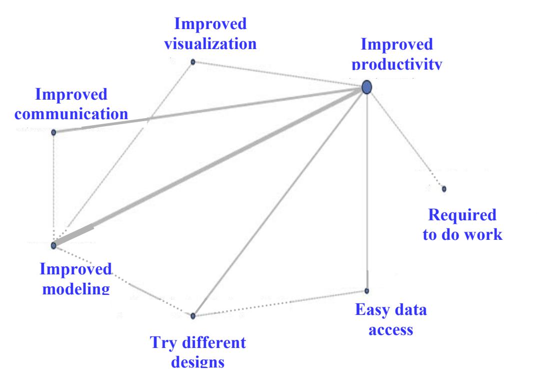
Some examples that illustrate the types of mission-critical software that were grouped under the different categories are

- Modeling: Pro-Engineer CAD, ASPEN, Unigraphics, HEC-RAS
- Project management: Microsoft Project, Field Manager and others.
- Industry-specific: Includes software that is specific for a certain industry.
- Data management: Statistical software, Data Analysis and Reporting, Oracle.
- Calculation tools: MATLAB, Simulink.

We asked how the mission-critical software listed in Table IV has changed the nature of their engineering practice. We grouped the responses into seven categories: 1) Improved productivity (34%); 2) Improved visualization (4%); 3) Improved communication (7.8%); 4) Improved modeling (13.8%); 5) Ability to try different designs (6.6%); 6) Easy data access (6%); 7) Required to perform daily work functions (3.3%). It is important to take into account that a single response can fall within two or more categories.

For deeper insight, we examined the relationships among the categories. For example, the majority of responses cite "improved productivity," but in what areas are engineers seeing productivity gains? Figure 3 shows the relationships.

FIGURE 3



THE RELATIONSHIPS AMONG CATEGORIES (IN BLUE) FOR THE RESPONSES ABOUT MISSION-CRITICAL SOFTWARE. THE THICKNESS OF THE LINES INDICATES THE STRENGTH OF THE RELATIONSHIP BETWEEN CATEGORIES; THE SIZE OF THE OVALS RELATES TO THE NUMBER OF RESPONSES IN EACH CATEGORY.

Most engineers see the improvements in modeling, visualization, and the support for trying different designs / "what if" analyses as the primary areas in which they see productivity gains.

COMPUTATIONAL THINKING A FOUNDATION FOR CURRICULAR REVISION

Since the particular computational skills identified from our interviews and surveys can be idiomatic to particular disciplines, industries, or even companies, it is important to find common threads that can be used to guide curriculum design. To help us identify and organize the common threads, we draw on Wing's exemplars of "computational thinking" (CT) [2] as listed below:

- Builds on power and limits of computing processes
- Involves solving problems, designing systems, and understanding human behavior
- Reformulating a difficult problem into one we can solve
- Thinking recursively
- Using abstraction and decomposition
- Thinking in terms of prevention, protection and recovery from worst case scenarios
- Using heuristic reasoning to discover a solution
- Complements and combines math and engineering thinking

Our task is not defined as using *all* the exemplars of CT activities listed above, but rather around using only those that are relevant in engineering contexts as informed by our industrial feedback.

After aligning the skills enumerated by our industrial partners with these exemplars of Computational Thinking, we have a complete "map" of the key computational problem solving components that are driven by industrial feedback and are relevant in an engineering context.

At the time of this writing we are still analyzing our data and have begun to map the skills onto the principles. Preliminary results show that computation in engineering professions draw from most exemplar CT as listed above, but with computation and automation at the forefront as more and more engineering knowledge is encapsulated in more sophisticated software. This preliminary map is summarized in Table VI.

TABLE VI
INTEGRATION OF ENGINEERING WORKFORCE SKILL-SET AND EXEMPLARS OF COMPUTATIONAL THINKING

Principle	Skill Set
Problem solving, limits of computing processes	Calculation tools, programming, excel, data management, mathematical analysis software, verifying correctness of computer results
Prevention, protection and recovery from worst case scenarios	Program logic controller, modeling and simulation software
Human factors	Interface between human and processes
Using abstraction and decomposition	CAD software, visualization software

The next step in the process is to use these concepts to find opportunities for curricular integration. Computation for engineering cannot simply be addressed with one or two courses in computing, but must be integrated as part of an engineer's training to become a "Holistic Engineer." We will introduce a series of authentic engineering problems derived from industry that require the use of various computing concepts for their solution. These problems will be part of the disciplinary courses (i.e., Chemical Engineering, Civil Engineering, etc.) but will be developed in consultation with stakeholders from industry, employees, and faculty from engineering disciplines and computer science to ensure that the problems are representative of engineering practice, disciplinary context, and computing concepts.

To accomplish this, we infuse Wing's exemplars of "computational thinking" across all four years of the engineering curriculum [5]. With help from stakeholders, we will identify problems from industry that are appropriate to a variety of courses and could be used with varying degrees of complexity depending upon the course level. First-year courses would use simplified versions of problems. As students progress through their programs, the problems will become more complex. However, the underlying computing concepts will be addressed across the various courses and throughout the degree program. For example, in Chemical Engineering concepts may be distributed as indicated in Table VII.

TABLE VII
CURRICULAR INTEGRATION

	Concept 1	Concept 2	Concept 3	Concept …	Concept N
Year 1	Intro to Eng.		Intro to Eng		Intro to Eng
Year 2	Conservation Principles	Applied math in ChE		Applied math in ChE	Applied math in ChE
Year 3	Separations		Separations	Separations	
Year 4	Proc Control & Modeling	Proc. Control & Modeling	Design	Proc. Control & Modeling	Design

SUMMARY AND FUTURE DIRECTIONS

Noting the strong demand for the preparation of a globally competitive U.S. workforce with knowledge and understanding of critical computing concepts, the CPACE initiative places primary emphasis on the engagement of stakeholders from multiple sectors to identify engineering computational problem-solving skills, to define how these skills can be integrated across curricula, and to revise the curricula to integrate computational problem-solving directly informed by industry needs.

To meet this vision, we brought together faculty and administrators from Michigan State University (MSU), Lansing Community College (LCC) and – through the

Corporation for Skilled Workforce (CSW) – leaders from business, industry and professional organizations in the region, who have an interest in transforming undergraduate computing education. We created and implemented a highly collaborative process to engage these participants in dialogue, explore common interests, and identify promising practices around computing knowledge and skills for the engineering workforce [3].

Using survey and interview instruments we determined stakeholders' engineering use of computer technology and the computational skills needed in their businesses. Our findings on the general engineering themes are consistent with other research on engineering education [1, 6]: a) employers place a high value on interpersonal skills such as communication, ability to organize and present data, and the ability to function in a team; b) employers see critical and innovative thinking and problem solving as important attributes; and c) employers see trends towards computational globalization which translates to the need for engineers to understand business practices and the importance of integrating engineering data across larger systems. Employers place a high value on the ability of engineers to understand both engineering principles and computational principles that allow them to use computational tools to solve engineering problems by moving between physical systems and abstractions in software.

We are aligning the computational problem solving threads that incorporate the key components of computation in the engineering workplace--as identified by our employer interviews and employee survey data--with fundamental computer science concepts as expressed in the exemplars for Computational Thinking (Figure 1- Phase II; node III). This will serve as a framework for discussion and course redesign.

We are developing a proposal to implement this integrated curriculum at Michigan State University and Lansing Community College. Key to success of this project will be faculty acceptance of curricular change and collaboration among the stakeholders from all sectors (Figure 1) to identify industrial problems that can be used throughout the curriculum in different disciplines.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under CPATH award 0722221. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

REFERENCES

[1] Educating Engineers: Designing for the future of the field. The Carnegie Foundation for the Advancement of Teaching 2008

[2] Jeanette Wing: Computational Thinking, Communications of the ACM March 2006/Vol. 49, No. 3, p 33

[3] ASEE Vergara et al. (2009). Leveraging workforce needs to inform curricular change in computing education for engineering: The CPACE project. Paper presented at the ASEE Conference and Exposition, Austin, TX.

[4] SPSS Text Analysis for Surveys, v 3.0, SPSS, Chicago, IL.

[5] Ohmann, P. R., Green, A. S., & Johnston, M. E. (2008). Infusing computation throughout the undergraduate curriculum. Computers in Education Journal, 18 (3), 12-22.

[6] Committee on the Engineer of 2020, Educating the Engineer of 2020: Adapting Engineering Education to the New Century. National Academy Press: Washington, DC, 2005.

AUTHOR INFORMATION

Claudia E. Vergara. Research Associate. Center for Research on College Science Teaching and Learning. MSU

Mark Urban-Lurain. Director of Instructional Technology Research & Development Division of Science and Mathematics Education College of Natural Science. MSU

Cindee Dresen. Director of Talent Development. CSW

Tammy Coxen. Policy Associate. CSW

Taryn MacFarlane. Policy Associate. CSW

Daina Briedis. Associate Professor Chemical Engineering and Materials Science. MSU

Neeraj Buch. Professor Civil and Environmental Engineering. MSU

Abdol-Hossein Esfahanian. Associate Chairperson Computer Science and Engineering, MSU.

Jeannine LaPrad. President and CEO. CSW

Louise Paquette. Professor. Math and Computer Science Department. LCC

Jon Sticklen. Director Applied Engineering Sciences. College Coordinator for Engineering Education Research. Associate Professor of Computer Science. MSU

Kysha Frazier Senior Policy Associate. CSW

Thomas F. Wolff. Associate Dean for Undergraduate Studies, College of Engineering, MSU. Project Principal Investigator.