Launching Curricular Guidelines for Computer Engineering: CE2016

John Impagliazzo
Engineering and Applied Science
Hofstra University
Hempstead, New York USA
john.impagliazzo@hofstra.edu

Susan Conry
Electrical and Computer Engineering
Clarkson University
Potsdam, New York, USA
sconry@clarkson.edu

Eric Durant
Electrical Engineering and Computer Science
Milwaukee School of Engineering
Milwaukee, Wisconsin USA
durant@msoe.edu

Joseph L. Hughes
Electrical and Computer Engineering
Georgia Tech
Atlanta, Georgia USA
joe.hughes@ece.gatech.edu

Russ Meier
Electrical Engineering & Computer Science
Milwaukee School of Engineering
Milwaukee, Wisconsin USA
meier@msoe.edu

***Abstract*—ACM and the IEEE Computer Society plan to release their computer engineering curriculum guidelines at the end of this calendar year. The curricular report, tagged CE2016, reflects the state-of-the-art in computer engineering education and practice that would be relevant for the coming decade. This panel presentation provides an overview of the report and it also provides unique perspectives from some steering committee members and other interested parties. The authors and participants will engage in discussions on ways to implement the guidelines to form new programs or to modify existing programs. The authors welcome all audience participation including overall comments and targeted editing assistance from the computer engineering education and industry communities.**

***Keywords*—Computer engineering; curriculum guidelines; CE2016; CE2004; ACM; IEEE Computer Society**

I. BACKGROUND

In 2011, the ACM and the IEEE Computer Society created the CE2004 Review Task Force and charged it with reviewing and determining the extent to which the document titled, "Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering" (CE2004) [1] required revision. After issuing survey invitations to over twenty thousand industry and academic constituents in the computer engineering field, it reported its findings to both societies. Part of the report included some new or expanded technical skill areas such as system on a chip (SoC) technologies, modern networking capabilities, software engineering agile methods and tools, embedded system design, parallel programming, and hardware/software co-design, and multicore architectures.

Based on the exploratory results, the two societies appointed a steering committee to develop a modern and meaningful update of the current report that was approaching a decade since its release. The new document would appropriately have to be forward looking given the continuing changes in the computer engineering landscape. Consequently, the steering committee produced new preliminary and interim reports with final publication scheduled for the end of 2016. The steering committee has sought input from computer engineering industrial and academic communities through surveys and by presenting workshops co-located with major conferences [e.g., 2-7].

II. RECOMMENDATION CONTENT

The knowledge areas in the body of knowledge (BOK) related to the computer engineering curricular guidelines appears in Table 1. Associated with each of these knowledge areas is a set of knowledge units that contain the learning outcomes expected of students by the time of graduation. As with all curricular guidelines, the knowledge areas are *not* courses. Hence, it is up to each program to align these knowledge areas and knowledge units with courses. The report provides some examples on ways to map knowledge units (and hence knowledge areas) to courses. See Appendix A for details.

The CE2016 report also addresses expected requirements in science and mathematics. Although one would expect a substantial amount of science in a computer engineering program, the report does not address which sciences are appropriate and leaves that decision to individual programs. With respect to mathematics, the report recommends at least discrete structures, analysis of continuous functions (calculus), probability and statistics, and linear algebra. See Appendix B for details.

TABLE 1. BODY OF KNOWLEDGE

Tag ID	Knowledge Area
CE-CAE	Circuits and Electronics
CE-CAL	Computing Algorithms
CE-CAO	Computer Architecture and Organization
CE-DIG	Digital Design
CE-ESY	Embedded Systems
CE-NWK	Computer Networks
CE-PPP	Preparation for Professional Practice
CE-SEC	Information Security
CE-SGP	Signal Processing
CE-SPE	Systems and Project Engineering
CE-SRM	Systems Resource Management
CE-SWD	Software Design

With respect to laboratory experiences, the report provides suggestions on ways to integrate such experiences within the curriculum. It also provides suggestions for laboratory configurations appropriate for computer engineering programs.

III. Presentation and Presenters

This CE2016 presentation encourages discussion and provides a platform for inquiry relative to the CE2016 report. Its goals include the following.

- Present a high-level summary of the CE2016 report;
- Show ways in which the CE body of knowledge transforms into courses to form a curriculum;
- Provide alternate perspectives toward curricular implementation for programs with alternate missions;
- Engage in a robust discussion between panel members and the audience.

The presenters on this panel appear as follows together with their perspectives on the discussion.

John Impagliazzo (Hofstra University) is chair of the CE2016 steering committee; he was a member of the CE2004 committee and a principal co-author of that report. He is also a member of a committee to a parallel project for information technology (IT2017). John is moderator of the panel and he will present a brief overview of the CE2016 document and how it evolved over recent years.

Susan Conry (Clarkson University) is a member of the steering committee and was the past chair of the Engineering Accreditation Commission (EAC) of ABET. Susan will discuss ways in which the computer engineering curriculum complements the current EAC/ABET criteria and the proposed EAC criteria. She will highlight ways in which the CE2016 curricular guidelines satisfy those criteria.

Eric Durant (Milwaukee School of Engineering) is a member of the steering committee and he has taken leadership roles on behalf of the committee at conferences and in publications. Eric will summarize the body of knowledge of the report and explain reasons for its current form.

Joseph L. Hughes (Georgia Tech) has a long history of computer engineering education is an active member of the steering committee, and was a member of the CE2004 committee. Joe usually provides unique insights on the development of the report and he will summarize ways in which institutions can transform their courses to conform with the body of knowledge.

Russ Meier (Milwaukee School of Engineering) is a strong and long-time proponent of computer engineering education and he has been an active participant in dialogue during earlier steering committee presentations. Russ will articulate practical elements toward the implementation of the CE2016 guidelines.

IV. Format and Audience

Because the CE2016 project is coming to a close and because there are diverse opinions on the related report, it is important to provide a forum for discussion. This penultimate event is better served as a panel rather than a workshop or other setting. Panelists will present diverse views on the subject and audience participation is essential. This presentation plans to engage computer engineering educators and practitioners toward implementing the tenets of this report.

References

[1] D. Soldan *et al.* “Curriculum guidelines for undergraduate degree programs in computer engineering,” (CE2004). December 12, 2004. Retrieved April 14, 2013 from http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf

[2] J. Impagliazzo, S. Conry, E. Durant, A. McGettrick, T. Wilson, and M. Thornton, “Special session: computer engineering review task force report,” ACM Special Interest Group on Computer Science Education (SIGCSE) Conference, March 2, 2012, Raleigh, NC.

[3] E. Durant, J. Impagliazzo, S. Conry, A. McGettrick, M. Thornton, and T. Wilson, “Special session: CE2004 revisions (computer engineering curriculum guidelines),” Frontiers in Education (FIE) Conference, October 6, 2012, Seattle, WA.

[4] E. Durant, J. Impagliazzo, S. Conry, A. McGettrick, M. Thornton, and T. Wilson, “Pre-conference workshop: Computer engineering curriculum guidelines,” Frontiers in Education (FIE) Conference, October 23, 2013, Oklahoma City, OK.

[5] E. Durant, J. Impagliazzo, S. Conry, R. Reese, M. Thornton, H. Lam, and V. Nelson, “Setting the stage for CE2016: A revised body of knowledge,” Frontiers in Education (FIE) Conference, October 22, 2014, Madrid, Spain.

[6] J. Impagliazzo and E. Durant, “Toward a modern curriculum for computer engineering,” IEEE International Conference on Teaching, Assessment and Learning (TALE), December 8, 2014, Wellington, New Zealand.

[7] E. Durant, J. Impagliazzo, S. Conry, R. Reese, H. Lam, V. Nelson, and J. Hughes, “CE2016 Update (Panel Discussion),” American Society for Engineering Education (ASEE) Annual Conference, June 16, 2015, Seattle, WA.

Appendix A

CE2016 Body of Knowledge
(CE Core Hours: 420)

Knowledge Areas and Knowledge Units			
CE-CAE	**Circuits and Electronics** [50 core hours]	**CE-CAL**	**Computing Algorithms** [30 core hours]
CE-CAE-1	History and overview [1]	CE-CAL-1	History and overview [1]
CE-CAE-2	Relevant tools, standards, and/or engineering constraints [3]	CE-CAL-2	Relevant tools, standards and/or engineering constraints [1]
CE-CAE-3	Electrical quantities and basic elements [4]	CE-CAL-3	Basic algorithmic analysis [4]
CE-CAE-4	Electrical circuits [11]	CE-CAL-4	Algorithmic strategies [6]
CE-CAE-5	Electronic materials, diodes, and bipolar transistors [7]	CE-CAL-5	Classic algorithms for common tasks [3]
CE-CAE-6	MOS transistor circuits, timing, and power [12]	CE-CAL-6	Analysis and design of application-specific algorithms [6]
CE-CAE-7	Storage cell architecture [3]	CE-CAL-7	Parallel algorithms and multi-threading [6]
CE-CAE-8	Interfacing logic families [3]	CE-CAL-8	Algorithmic complexity [3]
CE-CAE-9	Operational amplifiers [3]	CE-CAL-9	Scheduling algorithms
CE-CAE-10	Mixed-signal circuit design [3]	CE-CAL-10	Basic computability theory
CE-CAE-11	Design parameters and issues		
CE-CAE-12	Circuit modeling and simulation methods		
CE-CAO	**Computer Architecture and Organization** [60 core hours]	**CE-DIG**	**Digital Design** [50 core hours]
CE-CAO-1	History and overview [1]	CE-DIG-1	History and overview [1]
CE-CAO-2	Relevant tools, standards and/or engineering constraints [1]	CE-DIG-2	Relevant tools, standards, and/or engineering constraints [2]
CE-CAO-3	Instruction set architecture [10]	CE-DIG-3	Number systems and data encoding [3]
CE-CAO-4	Measuring performance [3]	CE-DIG-4	Boolean algebra applications [3]
CE-CAO-5	Computer arithmetic [3]	CE-DIG-5	Basic logic circuits [6]
CE-CAO-6	Processor organization [10]	CE-DIG-6	Modular design of combinational circuits [8]
CE-CAO-7	Memory system organization and architectures [9]	CE-DIG-7	Modular design of sequential circuits [9]
CE-CAO-8	Input/Output interfacing and communication [7]	CE-DIG-8	Control and datapath design [9]
CE-CAO-9	Peripheral subsystems [7]	CE-DIG-9	Design with programmable logic [4]
CE-CAO-10	Multi/Many-core architectures [5]	CE-DIG-10	System design constraints [5]
CE-CAO-11	Distributed system architectures [4]	CE-DIG-11	Fault models, testing, and design for testability
CE-ESY	**Embedded Systems** [40 core hours]	**CE-NWK**	**Computer Networks** [20 core hours]
CE-ESY-1	History and overview [1]	CE-NWK-1	History and overview [1]
CE-ESY-2	Relevant tools, standards, and/or engineering constraints [2]	CE-NWK-2	Relevant tools, standards, and/or engineering constraints [1]
CE-ESY-3	Characteristics of embedded systems [2]	CE-NWK-3	Network architecture [4]
CE-ESY-4	Basic software techniques for embedded applications [3]	CE-NWK-4	Local and wide area networks [4]
CE-ESY-5	Parallel input and output [3]	CE-NWK-5	Wireless and mobile networks [2]
CE-ESY-6	Asynchronous and synchronous serial communication [6]	CE-NWK-6	Network protocols [3]
CE-ESY-7	Periodic interrupts, waveform generation, time measurement [3]	CE-NWK-7	Network applications [2]
CE-ESY-8	Data acquisition, control, sensors, actuators [4]	CE-NWK-8	Network management [3]
CE-ESY-9	Implementation strategies for complex embedded systems [7]	CE-NWK-9	Data communications
CE-ESY-10	Techniques for low-power operation [3]	CE-NWK-10	Performance evaluation
CE-ESY-11	Mobile and networked embedded systems [3]	CE-NWK-11	Wireless sensor networks
CE-ESY-12	Advanced input/output topics [3]		
CE-ESY-13	Computing platforms for embedded systems		
CE-PPP	**Preparation for Professional Practice** [20 core hours]	**CE-SEC**	**Information Security** [20 core hours]
CE-PPP-1	History and overview [1]	CE-SEC-1	History and overview [2]
CE-PPP-2	Relevant tools, standards, and/or engineering constraints [1]	CE-SEC-2	Relevant tools, standards, and/or engineering constraints [2]
CE-PPP-3	Effective communication strategies [1]	CE-SEC-3	Data security and integrity [1]
CE-PPP-4	Multidisciplinary team approaches [1]	CE-SEC-4	Vulnerabilities and exploitation [4]
CE-PPP-5	Philosophical frameworks and cultural issues [2]	CE-SEC-5	Resource protection models [1]
CE-PPP-6	Engineering solutions and societal effects [2]	CE-SEC-6	Secret and public key cryptography [3]
CE-PPP-7	Professional and ethical responsibilities [4]	CE-SEC-7	Message authentication codes [1]
CE-PPP-8	Intellectual property and legal issues [3]	CE-SEC-8	Network and web security [3]
CE-PPP-9	Contemporary issues [2]	CE-SEC-9	Authentication [1]
CE-PPP-10	Business and management issues [3]	CE-SEC-10	Trusted computing [1]
CE-PPP-11	Tradeoffs in professional practice	CE-SEC-11	Side-channel attacks [1]

Knowledge Areas and Knowledge Units	
CE-SGP **Signal Processing** [30 core hours] CE-SGP-1 History and overview [1] CE-SGP-2 Relevant tools, standards, and/or engineering constraints [3] CE-SGP-3 Convolution [3] CE-SGP-4 Transform analysis [5] CE-SGP-5 Frequency response [5] CE-SGP-6 Sampling and aliasing [3] CE-SGP-7 Digital spectra and discrete transforms [6] CE-SGP-8 Finite and infinite impulse response filter design [4] CE-SGP-9 Window functions CE-SGP-10 Multimedia processing	**CE-SPE** **Systems and Project Engineering** [35 core hours] CE-SPE-1 History and overview [1] CE-SPE-2 Relevant tools, standards and/or engineering constraints [3] CE-SPE-3 Project management principles [3] CE-SPE-4 User Experience / Human-computer interaction [6] CE-SPE-5 Risk, dependability, safety and fault tolerance [3] CE-SPE-6 Hardware and software processes [3] CE-SPE-7 Requirements analysis and elicitation [2] CE-SPE-8 System specifications [2] CE-SPE-9 System architectural design and evaluation [4] CE-SPE-10 Concurrent hardware and software design [3] CE-SPE-11 System integration, testing and validation [3] CE-SPE-12 Maintainability, sustainability, manufacturability [2]
CE-SRM **Systems Resource Management** [20 core hours] CE-SRM-1 History and overview [1] CE-SRM-2 Relevant tools, standards, and/or engineering constraints [1] CE-SRM-3 Managing system resources [8] CE-SRM-4 Real-time operating system design [4] CE-SRM-5 Operating systems for mobile devices [3] CE-SRM-6 Support for concurrent processing [3] CE-SRM-7 System performance evaluation CE-SRM-8 Support for virtualization	**CE-SWD** **Software Design** [45 core hours] CE-SWD-1 History and overview [1] CE-SWD-2 Relevant tools, standards, and/or engineering constraints [3] CE-SWD-3 Programming constructs and paradigms [12] CE-SWD-4 Problem-solving strategies [5] CE-SWD-5 Data structures [5] CE-SWD-6 Recursion [3] CE-SWD-7 Object-oriented design [4] CE-SWD-8 Software testing and quality [5] CE-SWD-9 Data modeling [2] CE-SWD-10 Database systems [3] CE-SWD-11 Event-driven and concurrent programming [2] CE-SWD-12 Using application programming interfaces CE-SWD-13 Data mining CE-SWD-14 Data visualization

Appendix B

Related CE Mathematics
(120 Core Hours)

Mathematics Knowledge Areas and Units	
CE-ACF **Analysis of Continuous Functions** [30 core hours] CE-ACF-1 History and overview [1] CE-ACF-2 Relevant tools and engineering applications [1] CE-ACF-3 Differentiation methods [4] CE-ACF-4 Integration methods [6] CE-ACF-5 Linear differential equations [8] CE-ACF-6 Non-linear differential equations [3] CE-ACF-7 Partial differential equations [5] CE-ACF-8 Functional series [2]	**CE-DSC** **Discrete Structures** [30 core hours] CE-DSC-1 History and overview [1] CE-DSC-2 Relevant tools and engineering applications [1] CE-DSC-3 Functions, relations, and sets [6] CE-DSC-4 Boolean algebra principles [4] CE-DSC-5 First-order logic [6] CE-DSC-6 Proof techniques [6] CE-DSC-7 Basics of counting [2] CE-DSC-8 Graph and tree representations and properties [2] CE-DSC-9 Iteration and recursion [2]
CE-LAL **Linear Algebra** [30 core hours] CE-LAL-1 History and overview [1] CE-LAL-2 Relevant tools and engineering applications [2] CE-LAL-3 Bases, vector spaces, and orthogonality [4] CE-LAL-4 Matrix representations of linear systems [4] CE-LAL-5 Matrix inversion [2] CE-LAL-6 Linear transformations [3] CE-LAL-7 Solution of linear systems [3] CE-LAL-8 Numerical solution of non-linear systems [4] CE-LAL-9 System transformations [3] CE-LAL-10 Eigensystems [4]	**CE-PRS** **Probability and Statistics** [30 core hours] CE-PRS-1 History and overview [1] CE-PRS-2 Relevant tools and engineering applications [2] CE-PRS-3 Discrete probability [5] CE-PRS-4 Continuous probability [4] CE-PRS-5 Expectation and deviation [2] CE-PRS-6 Stochastic Processes [4] CE-PRS-7 Sampling distributions [4] CE-PRS-8 Estimation [4] CE-PRS-9 Hypothesis tests [2] CE-PRS-10 Correlation and regression [2]