ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36817

Quantifying Compliance of Computer Engineering Curricula with IEEE and ACM Guidelines

Filip Cuckov

Dr. Filip Čučkov is an Assistant Professor in the Electrical and Computer Engineering Program at Wentworth Institute of Technology (WIT). His expertise is in designing embedded computing architectures for Cyber-Physical Systems (CPS) and developing technologies and engineering solutions for robotics, automated systems, and human-in-the-loop CPS. He is passionate about engineering education and integrating students into his research program. Dr. Čučkov is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), and a member of the American Society for Engineering Education (ASEE), and the Project Management Institute (PMI).

Marisha Rawlins

Wayne Bynoe (Assistant Professor)

James R Mccusker (Associate Professor)

Quantifying Compliance of Computer Engineering Curricula with IEEE and ACM Guidelines

Abstract

This paper presents a methodology, applied via a novel framework, that quantifies the degree to which computer engineering curricula meet the most current guidelines prescribed by the IEEE and ACM. By using our framework, engineering educators can identify deficiencies, overlap, and excess in their computer engineering curricula. This information can support the broader engineering education community's future curriculum reimagination and redesign process. We detail the procedure centered around a self-audit and share an open-sourced framework for its execution. We also present sample data obtained by executing the framework on our computer engineering curriculum that visually quantify the overall compliance of computer engineering curricula to IEEE/ACM recommendations and further detail the compliance with each of the computer engineering discipline's core knowledge areas. The feedback from the audit process and data from our presented framework are necessary to drive the reimagination and redesign process of future computer engineering curricula. Such curricula should meet the standards of today yet look forward to adapting to the guidelines of tomorrow, which are embodied by the IEEE/ACM Computing Curricula 2020 Paradigms for Global Computing Education.

Introduction

At our institution, like many others worldwide, it has been over a decade since we have reimagined and redesigned our engineering curricula. Since then, we have ensured and confirmed compliance with accreditation agencies [1], perfected the delivery of courses, and assessed learning outcomes to ensure that our graduates can be successful in all the different stages of their careers. The problem is that in the last ten years, the careers that await our graduates have changed fundamentally such that our curricula of today effectively do not prepare our students for the careers of tomorrow [2]. More importantly, the way students learn has also fundamentally changed [3], swiftly rendering our instructional methodologies obsolete. Our overarching motivation is to aid the engineering education community in the (re)design process of engineering curricula such that they can transcend the competency gap between the graduates of today and the careers of tomorrow.

The Joint Task Force on Computing Curricula by the Institute of Electrical and Electronics Engineers (IEEE) Computer Society and the Association of Computing Machinery (ACM) has published two Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering (CE), one in 2004 (CE2004) [4] and one in 2016 (CE2016) [5], that detail core Learning Outcomes (LO) of CE programs grouped in Knowledge Units (KU), typically mapped to classes and organized in Knowledge Areas (KA), that define the body of knowledge required by the discipline. Our framework is built on the CE2016 guidelines, which divide the CE body of knowledge into twelve core KAs, all together composed of one hundred and thirty-five KUs, further detailed by a total of nine hundred and eight LOs, one hundred and ten of which are declared as elective, thus leaving seven hundred and ninety-eight LOs considered core to the CE discipline. We are aware that the 908 LOs generated directly from the CE2016 guidelines may not be independent, but we chose to present results in this study based on the raw outcomes for simplicity. The data visualization framework presented remains valid. This matter may be addressed in a future study.

Looking ahead beyond 2020 [3], in the ACM/IEEE Computing Curricula 2020 Paradigms for Global Computing Education (CC2020) [6], the joint task force of these professional associations does not prescribe specific CE curricula but rather defines the core competencies of graduates from next-generation computing programs. CC2020 details thirty-four competencies for CE graduates that are streamlined into six knowledge areas: Hardware, Systems Architecture and Infrastructure, Software Fundamentals, Software Development, Systems Modeling, and Systems and Organizations; a welcome departure from the over-defined KA/KU/LO prescribed by the CE2016 guidelines.

Automated tools for assessment of competencies [7] and mapping the same to a curriculum have been widely reported in engineering education literature [8], although the research has been generally applied to engineering as a whole rather than being focused on program-specific competencies [9]. The framework presented in this paper fills a portion of that gap, specifically for CE curricula. Diagnostic tools for assessing competencies [10] applied to existing curricula can be very helpful in detecting inefficiencies and addressing them to improve mapping adequacies in further revisions, and can aid in reducing a curriculum's complexity to ensure it is transfer-friendly [11], while also incorporating the desired competencies of industry 4.0 [2]. Considerations for globalization and workforce mobility are also essential in defining CE competencies, supporting an international understanding of what the CE field encompasses [12], as well as what CE competencies are required in worldwide industries [13].

Once curriculum (re)designers have insight into the degree to which their curriculum meets the latest IEEE/ACM guidelines by executing the framework presented in this paper, the next step would be to map via a novel unified framework [14] using an analytical hierarchical-based process [15], the CE2016 12 KAs, 135 KUs, and 908 LOs to the 34 competencies in 6 areas defined in CC2020, therefore assuring forward compliance of their curricula and meeting the demand of the industries of tomorrow.

Framework and Methods

The framework for quantifying the compliance of CE curricula with 2016 IEEE/ACM recommendations consists of 12 independent self-assessments based on the CE core KAs: CAE - Circuits and Electronics, CAL - Computing Algorithms, CAO - Computer Architecture and Organization, DIG - Digital Design, ESY - Embedded Systems, NWK - Computer Networks, PPP - Preparation for Professional Practice, SEC - Information Security, SGP - Signal Processing, SPE - Systems and Project Engineering, SRM - Systems Resource Management, and SWD - Software Design, that are then automatically analyzed to provide data and visualizations, as later demonstrated in the Results and Discussion section. The 12 KAs contain a total of 135 KUs, composed in total of 798 core LOs; while an additional 110 elective LOs are provided in the framework, they are considered optional and therefore not included in the analysis. It is important to note that the 798 core LOs are not evenly distributed across KAs, ranging from 32 at the lowest to 92 at the highest, with an average of 67 LOs per KA. Also, since the recommended time devoted to each KA varies from 20 to 60 hours, our framework provides a weighted compliance rating for each KA and the curriculum as a whole.

The self-audit process, described next, is executed using our framework to cross-examine a CE curriculum and identify if and where the core 798 LOs are covered. For each of the 135 KU, auditors must first identify which required classes in the CE curriculum touch upon the set of LOs defined in that KU. Table 1shows a sample mapping of the SWD KUs body of knowledge across 6 sample required courses.

Table 1. Sample mapping of the SWD KUs body of knowledge.

SWD KUs collectively encompass the following:	**Class 1**	**Class 2**	**Class 3**	**Class 4**
1. Programming paradigms and constructs	ECE 101	ECE 202	CS 110	N/A
2. Data structures and use of standard libraries	ECE 202	ECE 333	CS 220	N/A
3. Object-oriented design and modeling languages	ECE 333	ECE 471	CS 220	N/A
4. Testing and software quality concepts	CS 110	ECE 202	ECE 333	ECE 471
5. Tradeoffs among different software design methods	CS 220	ECE 471	N/A	N/A

Elective courses should not be considered as the material covered in them cannot be guaranteed to be delivered uniformly to all graduates of the program. To ensure data quality, ideally, the auditors should be faculty who have taught the identified classes (if not, then the appropriate faculty should be consulted) and should reference the respective syllabi to ensure the LOs are indeed addressed.

Auditors then proceed to mark each LO with a binary mark if it is covered in a particular class. In cases where a LO is optionally covered, auditors may choose to assign a probability weight (ranging from 0 to 1) instead of a binary mark (0 or 1) and may choose to apply a probabilistic model in the compliance analysis. The audit process results in a coverage matrix of each of the 135 KUs, as shown in Table 2 for example, revealing deficiencies (row 5), overlap (rows 1, 2, and 3), and excess (row 3).

The collection of 135 coverage matrices in 12 KAs is then automatically analyzed by our framework to give a visual representation on 13 radar plots of the degree to which a given KA (and, in turn, the collection of KAs that embody a CE curriculum) follows the 2016 IEEE/ACM guidelines.

Table 2. Sample coverage matrix for CAL - KU 4: Algorithmic Strategies.

CAL - KU 4: Algorithmic strategies	**ECE 101**	**ECE 202**	**CS 110**	**ECE 333**
1. Design and implement brute force algorithms.	✓		✓	
2. Design and implement greedy algorithms.		✓	✓	
3. Design an algorithm w/ a divide & conquer strategy.	✓	✓	✓	✓
4. Explain how recursive algorithms work.	✓			
5. Explain why heuristics are useful.				

Results and Discussion

We executed a self-audit of our CE curriculum using the presented framework and process described earlier. In this section, we present and analyze the sample data and visualizations that our framework provides. The data is visualized in 1 summary table (Table 3) and 13 radar plots, 1 for the overall compliance of the CE curriculum (Figure 1) showing the percentage of compliance in each of the 12 KAs, and 12 radar plots for each KA (Figures 2-13) detailing the rate of compliance with KUs contained in the respective KA. The radar plots are automatically generated by the framework in the following manner, using the DIG KA as an example. In the DIG KA there are 10 KUs: five with four LOs each, one with five LOs, one with six LOs, one with seven LOs, and two with nine LOs, for a total of 56 LOs. Auditors mark each LO with a binary coverage matrix, as illustrated in Table 2, and then the framework assigns a score of 1 to each LO that is covered at least once in the matrix, finally summing the LOs coverage and dividing by the total number of LOs in that KU to obtain the rate of compliance which is then mapped on the corresponding KA's radar plot.

Quantification of Overall Compliance

Figure 1 shows the overall compliance of our CE curriculum to the 2016 IEEE/ACM recommendations. As it can be seen in this sample visualization, overall, our CE curriculum complies with the recommendations to a high degree (82% total coverage), with all KAs above ~ 80%, except in the case of *Computer Architecture and Organization* (73%) and *Information Security* (41%). This is because our hardware security course is offered as an elective rather than a required course which would have increased this low compliance score. This type of visualization can be used to quickly identify knowledge areas for improvement, such as CAO and SPE, for example, as well as to comparatively analyze CE curricula from different institutions to determine focus, strengths, and weaknesses.

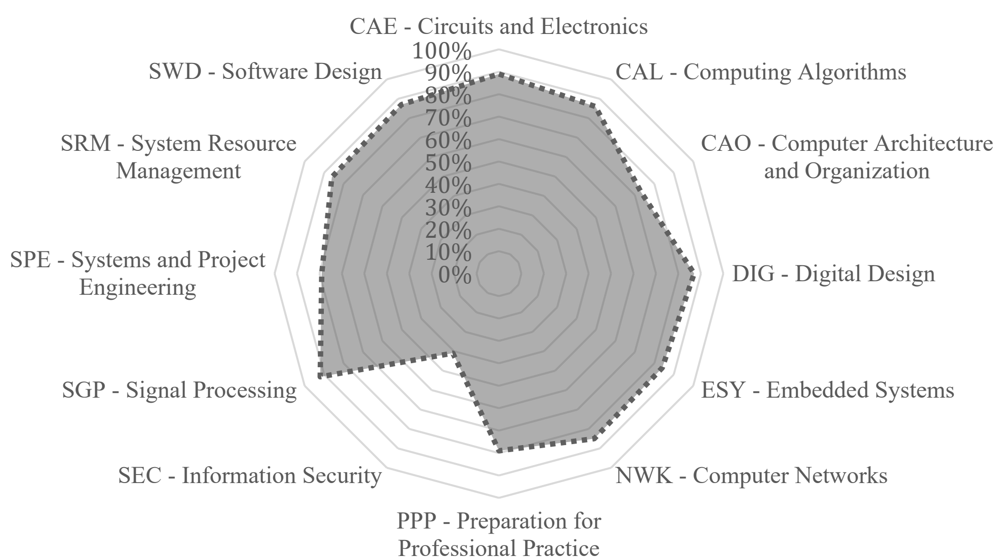

Figure 1. Overall compliance with computer engineering knowledge areas.

Our framework provides further detail to the compliance statistics, as shown in Table 3, where the number of recommended hours per KA is listed, number of KUs, and the number of LOs.

Table 3. Summary of compliance in each knowledge area.

Knowledge Area	Hrs	KUs	LOs	% Hrs	% LOs	LOs ✓	% ✓	Weighted % ✓
1. CAE - Circuits and Electronics	50	12	74	11%	9%	66	89%	10%
2. CAL - Computing Algorithms	30	10	32	7%	4%	27	84%	6%
3. CAO - Computer Arch. & Org.	60	11	75	14%	9%	55	73%	10%
4. DIG - Digital Design	50	11	56	11%	7%	49	88%	10%
5. ESY - Embedded Systems	40	13	62	9%	8%	52	84%	8%
6. NWK - Computer Networks	40	11	88	9%	11%	75	85%	8%
7. PPP - Prep. for Prof. Practice	20	11	92	5%	12%	73	79%	4%
8. SEC - Information Security	20	11	79	5%	10%	32	41%	2%
9. SGP - Signal Processing	30	11	53	7%	7%	49	92%	6%
10. SPE - Systems & Proj. Eng.	35	12	88	8%	11%	69	78%	6%
11. SRM - Sys. Resource Mgmt.	20	8	45	5%	6%	39	87%	4%
12. SWD - Software Design	45	14	54	10%	7%	47	87%	9%
Total	**440**	**135**	**798**	**100%**	**100%**	**633**	**79.32%**	**81.94%**

As it can be seen from the table, the 798 total LOs and 440 total hours spent on training students for competencies in LOs per KA are not equally distributed among the 12 KAs, ranging from 45 to 92 LOs, and from 20 to 60 hours, respectively. For example, faculty should spend 14% of the instructional time across the CE curriculum on CAO LOs, which contribute 9% to the total number of LOs, versus 5% of the time on SEC, which contains 10% of the total LOs. Therefore, each KA does not equally contribute to the overall compliance percentage and can be weighted based on these contributing factors. The raw average (non-weighted) compliance score of our CE curriculum would be 79.32%, but once weighing is considered, then the compliance is more-adequately computed to be 81.94%. It is important to note that our framework considers all KUs to contribute equally to a given KA, regardless of the number of LOs they may contain individually.

Quantification of Compliance in Knowledge Areas

The overall compliance visualization and data show only the KAs, and in turn, the following 12 figures provide further detail for each KA, showing the percentage of compliance with each of the KUs. This greater level of detail is critical for identifying possible improvements in KUs and could be used to categorize further the strengths and weaknesses of a given CE curriculum.

1. CAE – Circuits and Electronics

Figure 2 shows the detail of our CE curriculum's compliance in the CAE KA, with an overall 89% coverage indicating strength, while a closer inspection reveals KUs of concern, such as in the *mixed-signal circuit design* and *design parameters and issues* KUs. Curriculum designers could then reference the coverage matrix for a concern KU to identify the reasons behind the deficiency. For example, focusing on concerning LOs, within required courses in our CE curriculum, we don't devote much time to educating students to "analyze issues associated with the integration of digital and analog circuits in a single IC or package, including both benefits and challenges" and "provide examples of commercial mixed-signal devices", thus putting our compliance with the *mixed-signal circuit design* KU at 70%. Elective courses in the curriculum indeed train our students in these LOs, but we cannot count on them because we cannot guarantee that every CE student will take these elective courses.

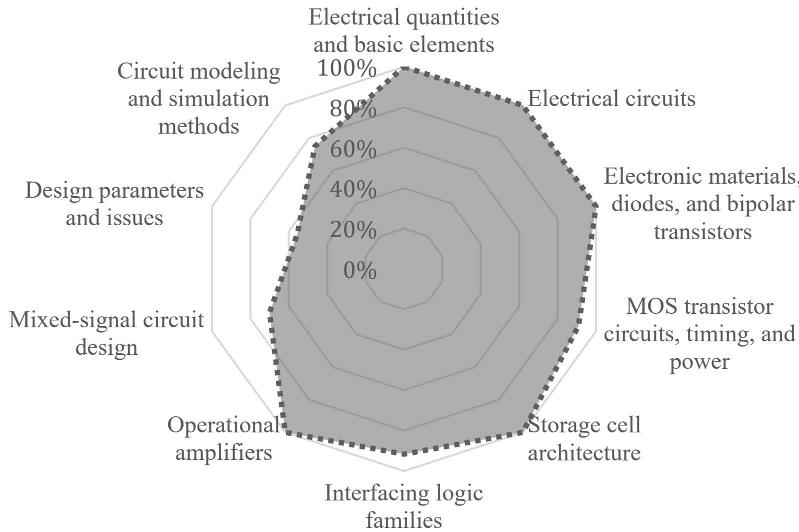

Figure 2. CAE – Circuits and Electronics compliance mapping (89% Overall).

2. CAL - Computing Algorithms

Figure 3 shows the compliance of our CE curriculum in the CAL KA, with an overall 84% coverage, indicating weakness in *analysis and design of application-specific algorithms*, and

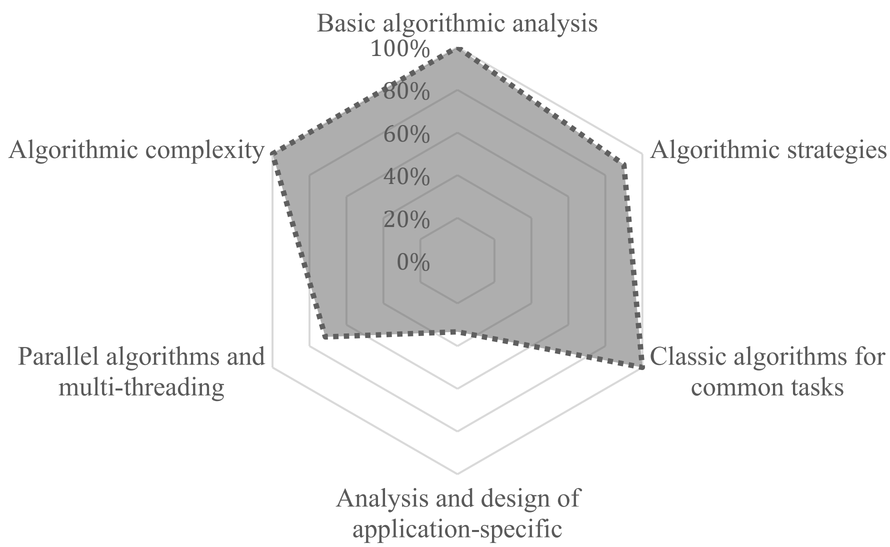

Figure 3. CAL - Computing Algorithms compliance mapping (84% Overall).

desired improvement in the *parallel algorithms and multi-threading* KUs. Further inspection of the LOs in that KU shows that this weakness is straight forward to address by ensuring that the 3 LOs, which are currently optionally or partially addressed, are covered in full in required courses.

3. CAO - Computing Architecture and Organization
Figure 4 shows the compliance of our CE curriculum in the CAO KA, with 73% overall coverage and notable KUs of concern in the *distributed system architectures* (17% compliance), and the *multi/many-core architectures* (58% compliance) KUs, which are covered in our elective advanced computer architecture course. While the *peripheral subsystems* and *input/output interfacing and communication* KUs are also lacking (with 29% and 50% compliance, respectively), upon further inspection of the 15 LOs it is apparent that five are not introduced at all in any required courses and doing so would improve our compliance rate.

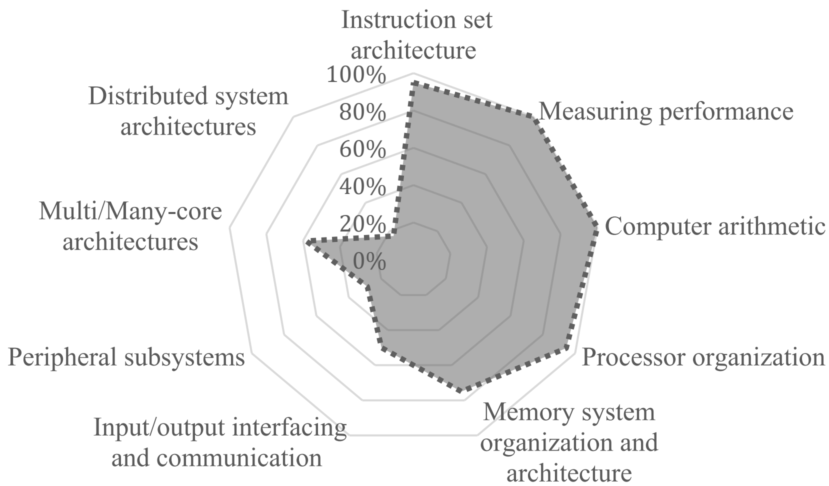

Figure 4. CAO - Computing Arch. and Org. compliance mapping (73% Overall).

4. DIG – Digital Design
Figure 5 shows the compliance of our CE curriculum in the DIG KA, with 88% overall coverage.

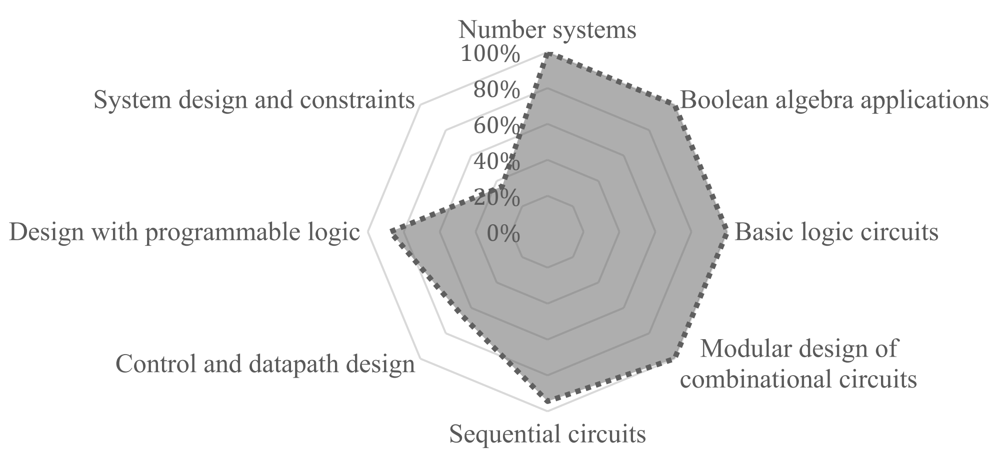

Figure 5. DIG – Digital Design compliance mapping (88% Overall).

The *system design and constraints coverage* KU's compliance of 36% indicates that more time should be devoted in class and laboratory to the synthesis of medium to large scale circuit design (also improving the *control and datapath design* KU with 67% compliance), which is a common problem in many other CE programs [16] that typically include synthesis techniques very late (as it is in our case) or at the very end of the semester in their digital design course.

5. ESY - Embedded Systems

The ESY KA compliance of our CE curriculum, with an overall coverage of 84%, is shown in Figure 6. The *advanced input/output*, *techniques for low-power operation*, and *data acquisition* KUs with 50%, %63, and 63% compliance, respectively, can be improved by focusing more on interfacing with various sensors and actuators via varying standardized interfaces.

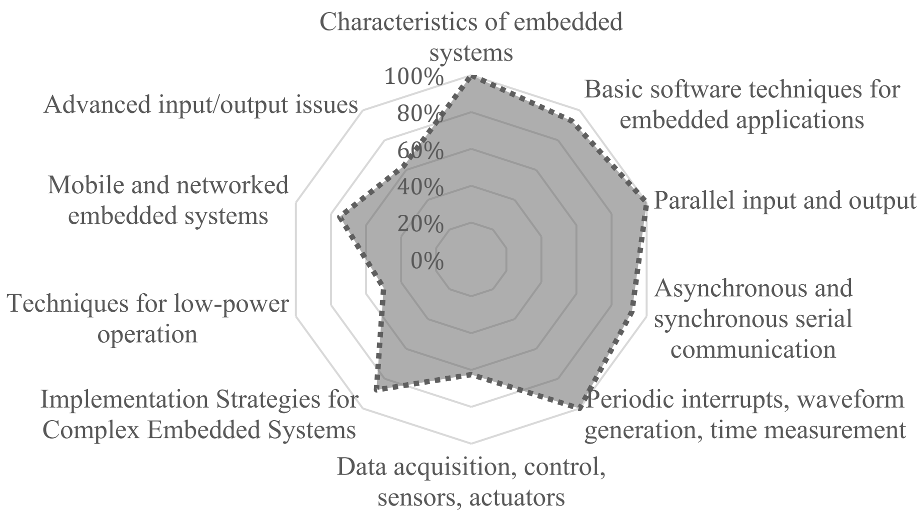

Figure 6. ESY - Embedded Systems compliance mapping (84% Overall).

6. NWK - Computer networks

The NWK KA complies with 85% of the IEEE/ACM recommendations of LOs, as shown in Figure 7, where a notable KU of concern is *Network management*. Upon closer inspection of the LOs, the following are missing in the coverage matrix of the KU:

- *Discuss four typical architectures for network management including the management console, aggregators, and device agents.*
- *Demonstrate the management of a device such as an enterprise switch through a management console.*
- *Contrast various network management techniques as they apply to wired and wireless networks such as topics on devices, users, quality of service, deployment, and configuration of these technologies;*

,thus, compliance would be increased by addressing these LOs in the required networking course.

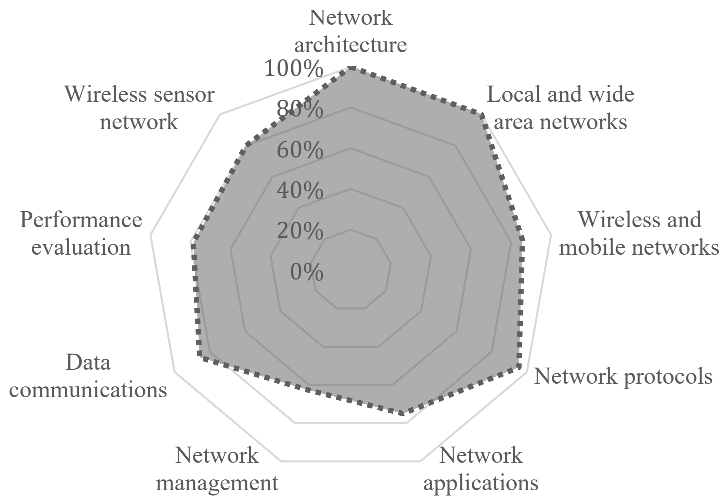

Figure 7. NWK - Computer Networks compliance mapping (85% Overall).

7. PPP - Preparation for Professional Practice

Upon closer inspection of the LOs in the PPP KA, which complies with 79% of the IEEE/ACM recommendations, as shown in Figure 8, a common weakness emerges in the *philosophical frameworks and cultural issues*, *contemporary issues*, and *professional and ethical responsibilities* KUs: ethics. Studies on how to better incorporate ethics education in CE curricula [17] suggest that the solution is threefold: develop case studies based on real-world examples for students to practice and develop ethical reasoning skills, combining (whenever possible) ethics with technical content across the curriculum, and engaging the faculty to actively take part in creating these activities and integrating them across the curriculum.

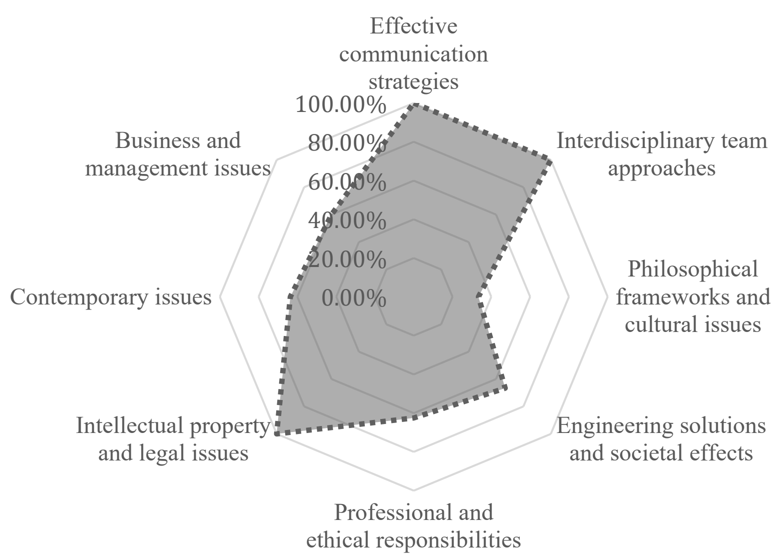

Figure 8. PPP - Preparation for Professional Practice compliance mapping (79% Overall).

8. SEC - Information Security

Our CE curriculum's overall compliance with the SEC KA is 41%, as shown in Figure 9, because our hardware security course, which covers all the prescribed KUs and LOs, is elective. A self-sufficient hardware security curriculum can be created free of advanced pre-requisites [18] since most cybersecurity is founded on the CE curriculum core LOs [19]. Thus, in our case, it makes sense to require our elective hardware security course, where the structural issue of how to make space for it in a crowded curriculum needs to be resolved.

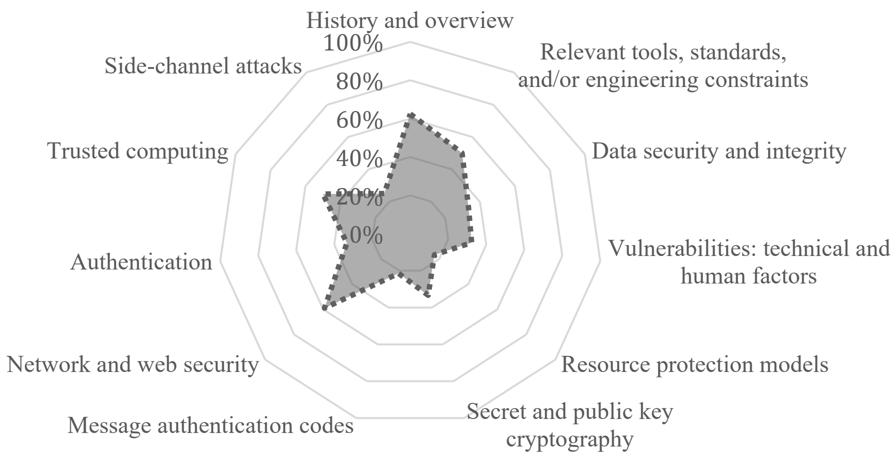

Figure 9. SEC - Information Security compliance mapping (41% Overall).

9. SGP - Signal Processing

Figure 10 shows that our CE curriculum is in 92% compliance with the IEEE/ACM recommendations in the SGP KA.

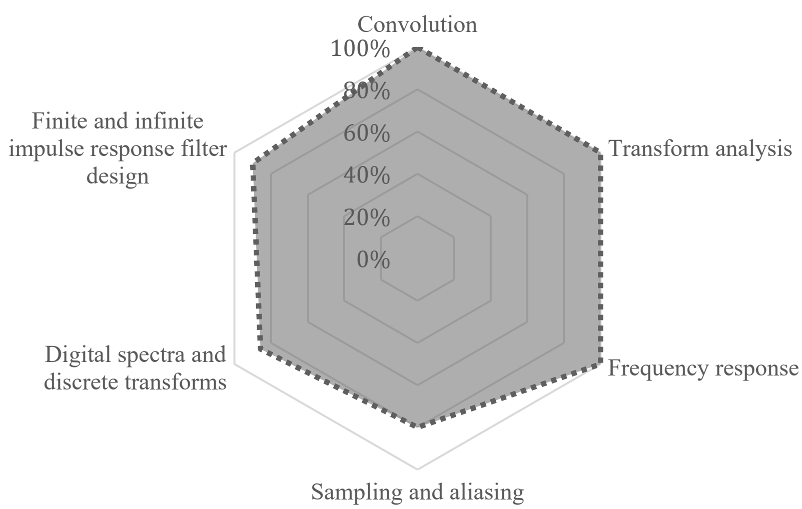

Figure 10. SGP - Signal Processing compliance mapping (92% Overall).

100% compliance in this KA would be ensured by addressing the following 3 LOs:

- *Calculate the errors or noise generated by sampling and quantizing*,
- *Explain how the block size controls the tradeoff between spectral resolution and density*,
- *Design a digital filter using analog techniques (e.g., bilinear transform) and explain its key parameters*.

10. SPE - Systems and Project Engineering

As Figure 11 shows, the 78% compliance in the SPE KA could be significantly improved by shifting our project-based *hardware-software co-design* [20] and *UI/UX* LOs in required courses.

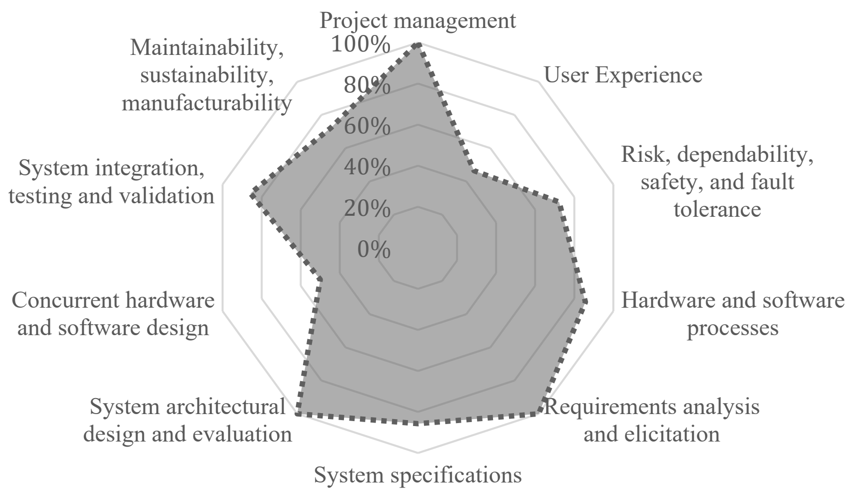

Figure 11. SPE - Systems and Project Engineering compliance mapping (78% Overall).

11. SRM - Systems Resource Management

Figure 12 shows the 87% overall compliance of our CE curriculum with the SRM KA. Areas for improvement include *support for visualization* and *operating systems for mobile devices*.

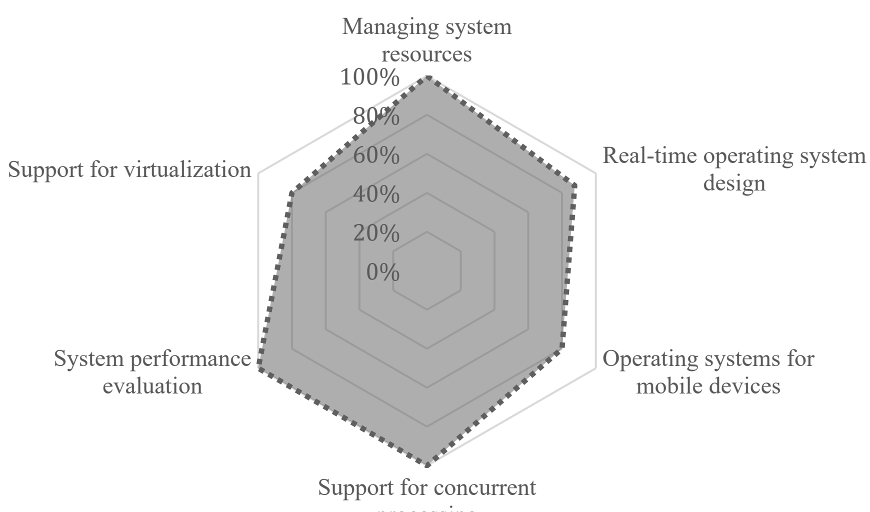

Figure 12. SRM - Systems Resource Management compliance mapping (87% Overall).

12. SWD - Software Design

Finally, Figure 13 shows that our CE curriculum complies 87% with IEEE/ACM recommendations in the SWD KA. Notable KUs for improvement include *event-driven and concurrent programming* and *problem-solving strategies*, which could be addressed by focusing on heuristics.

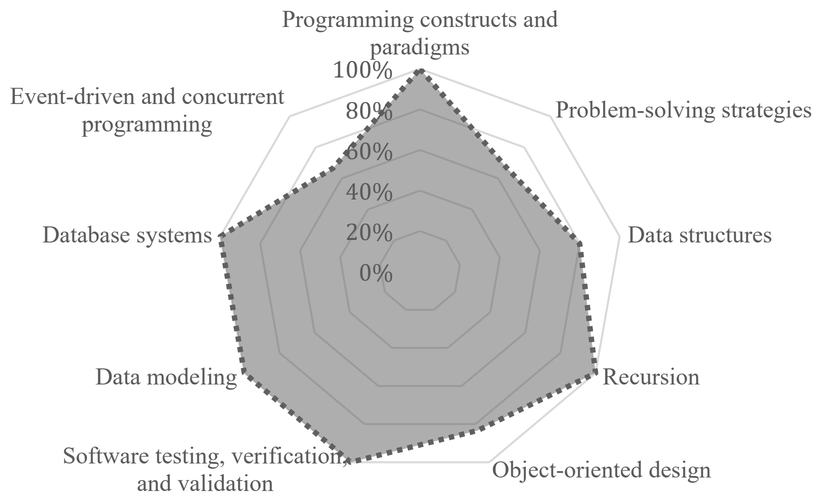

Figure 13. SWD - Software Design compliance mapping (87% Overall).

Conclusions

The framework for quantifying compliance of CE curricula with IEEE/ACM recommendations presented in this paper provides a structured approach to self-evaluation and presents valuable data visualizations that are necessary for a data-driven reimagination and redesign of future computer engineering curricula.

References

[1] "Criteria for Accrediting Engineering Programs, 2020 – 2021 | ABET." https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2020-2021/

[2] W. Maisiri and L. van Dyk, "Industry 4.0 Competence Maturity Model Design Requirements: A Systematic Mapping Review," in *2020 IFEES World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC)*, Nov. 2020, pp. 1–6. doi: 10.1109/WEEF-GEDC49885.2020.9293654.

[3] J. Qadir, K.-L. A. Yau, M. Ali Imran, and A. Al-Fuqaha, "Engineering Education, Moving into 2020s: Essential Competencies for Effective 21st Century Electrical and Computer Engineers," in *2020 IEEE Frontiers in Education Conference (FIE)*, Oct. 2020, pp. 1–9. doi: 10.1109/FIE44824.2020.9274067.

[4] "Curricula Recommendations." https://www.acm.org/education/curricula-recommendations

[5] "ce2016-final-report.pdf." Accessed: Feb. 04, 2022. [Online]. Available: https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf

[6] CC2020 Task Force, *Computing Curricula 2020: Paradigms for Global Computing Education*. New York, NY, USA: ACM, 2020. doi: 10.1145/3467967.

[7] P. Zhang and J. P. Wade, "Automated Assessment of Systems Engineering Competencies," presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018. Accessed: Feb. 04, 2022. [Online]. Available: https://peer.asee.org/automated-assessment-of-systems-engineering-competencies

[8] S. Arafeh, "Curriculum mapping in higher education: a case study and proposed content scope and sequence mapping tool," *J. Furth. High. Educ.*, vol. 40, no. 5, pp. 585–611, Sep. 2016, doi: 10.1080/0309877X.2014.1000278.

[9] D. W. Holmes, M. Sheehan, M. Birks, and J. Smithson, "Development of a competency mapping tool for undergraduate professional degree programmes, using mechanical engineering as a case study," *Eur. J. Eng. Educ.*, vol. 43, no. 1, pp. 126–143, Jan. 2018, doi: 10.1080/03043797.2017.1324404.

[10] G. Durand, C. Goutte, N. Belacel, Y. Bouslimani, and S. Léger, "A diagnostic tool for competency-based program engineering," in *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, New York, NY, USA, Mar. 2018, pp. 315–319. doi: 10.1145/3170358.3170402.

[11] D. Reeping, D. Grote, L. D. McNair, and T. Martin, "Curricular Complexity as a Metric to Forecast Issues with Transferring into a Redesigned Engineering Curriculum," presented at the 2020 ASEE Virtual Annual Conference Content Access, Jun. 2020. [Online]. Available: https://peer.asee.org/curricular-complexity-as-a-metric-to-forecast-issues-with-transferring-into-a-redesigned-engineering-curriculum

[12] A. Clear, J. Carter, G. Cross, A. Radenski, L. Tudor, and E. Tõnisson, "What's in a Name? International Interpretations of Computing Education Terminology," in *Proceedings of the 2015 ITiCSE on Working Group Reports*, 2015, pp. 173–186.

[13] A. Queiruga-Dios *et al.*, "Evaluating engineering competencies: A new paradigm," in *2018 IEEE Global Engineering Education Conference (EDUCON)*, Apr. 2018, pp. 2052–2055. doi: 10.1109/EDUCON.2018.8363490.

[14] F. Sánchez Carracedo *et al.*, “Competency Maps: An Effective Model to Integrate Professional Competencies Across a STEM Curriculum,” *J. Sci. Educ. Technol.*, vol. 27, no. 5, pp. 448–468, Oct. 2018, doi: 10.1007/s10956-018-9735-3.

[15] L. Lolong Lacatan and G. Mendoza Penuliar, “Competency-Based Mapping Tool in Personnel Management System using Analytical Hierarchy Process,” in *2021 The 4th International Conference on Machine Learning and Machine Intelligence*, New York, NY, USA, Sep. 2021, pp. 57–64. doi: 10.1145/3490725.3490734.

[16] H. A. Ochoa and M. V. Shirvaikar, “A Survey of Digital Systems Curriculum and Pedagogy in Electrical and Computer Engineering Programs,” presented at the 2018 Gulf Southwest Section Conference, Apr. 2019. Accessed: Feb. 04, 2022. [Online]. Available: https://peer.asee.org/a-survey-of-digital-systems-curriculum-and-pedagogy-in-electrical-and-computer-engineering-programs

[17] M. Ghorbani, A. A. Maciejewski, T. J. Siller, E. K. P. Chong, P. Omur-Ozbek, and R. A. Atadero, “Incorporating Ethics Education into an Electrical and Computer Engineering Undergraduate Program,” presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018. [Online]. Available: https://peer.asee.org/incorporating-ethics-education-into-an-electrical-and-computer-engineering-undergraduate-program

[18] A. Carpenter, “A Hardware Security Curriculum and its Use for Evaluation of Student Understanding of ECE Concepts,” presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018. [Online]. Available: https://peer.asee.org/a-hardware-security-curriculum-and-its-use-for-evaluation-of-student-understanding-of-ece-concepts

[19] D. W. Jacobson and J. A. Rursch, “Building a Cybersecurity Engineering Program? Begin by Cloning Your Computer Engineering Program,” presented at the 2020 ASEE Virtual Annual Conference Content Access, Jun. 2020. Accessed: Feb. 04, 2022. [Online]. Available: https://peer.asee.org/building-a-cybersecurity-engineering-program-begin-by-cloning-your-computer-engineering-program

[20] P. M. Matutino, T. Dias, and P. Sampaio, “Teaching Hardware/Software Co-Design Using a Project-Based Learning Strategy,” in *2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE)*, Jul. 2020, pp. 1–6. doi: 10.1109/TAEE46915.2020.9163759.