ORCHESTRATING CDIO-ALIGNED LEARNING TASKS THROUGH A COURSE-TYPE–DRIVEN ASSIGNMENT FRAMEWORK

Vairavel Gurusamy

Department of ECE, SRM Institute of Science and Technology, Tiruchirappalli Campus

Deiva Sundari P

Department of EEE, Easwari Engineering College

Siva Kumar M

Department of Mechanical Engineering, SRM TRP Engineering College

ABSTRACT

Engineering education is being increasingly challenged to equip graduates for a fast-changing technology thrust that demands implementation, systems thinking, and applied problem solving. Though the CDIO (Conceive–Design–Implement–Operate) framework provides a coherent basis for these challenges, faculty frequently report difficulty in translating these program-level principles into applicable course-level assignments. This paper addresses this gap by proposing a course-type-driven task design and orchestration framework. The framework is composed of three major contributions: a classification of nine undergraduate engineering course types; a CDIO-compliant task taxonomy of core, communication, and reflective tasks; and a logic for orchestrating student pathways under semester constraints. The tripartite task structure ensures that engineering action, communication, and judgment are interwoven in a single assignment. The framework was institutionally implemented in a third-year Mechanical Engineering course titled Design of Machine Elements (DME) as part of routine course delivery. By replacing existing numerical problem sets with an integrated design project, the framework activated prior learning and emulated professional work without transforming the curriculum. Results show that the framework increases assessment reliability, decreases faculty cognitive load in the design of assignments, and allows students to create engineering products of portfolio quality.

KEYWORDS

Engineering Education, CDIO Framework, Course-Level Implementation, Task Orchestration, Active Learning, Standards: 2, 3, 5, 7, 8 and 11.

INTRODUCTION

In the era of automation, biotechnology, cyber-physical systems, electric mobility and artificial intelligence, the engineering education sector has increasingly appreciated the significance of preparing engineering graduates for the rapidly evolving technological environment. National Academy of Engineering (2020) foresees these advances to transform the way engineering practice is performed across the globe and will compel educational institutions to rethink graduate education to prepare them for exciting yet challenging real-world engineering problems. Today's hiring managers expect graduates to be project-ready, i.e., not only having conceptual knowledge but capable of system thinking, applied problem solving, design judgment and execution. UNESCO (2021) believes society will also need engineers capable of designing ethical, socially responsible and sustainable solutions.

Undergraduate engineering programs also accommodate students whose professional paths become increasingly diverse. Placement-oriented students want to demonstrate skill profiles based on portfolio-level tasks. Research-oriented students preparing for MS or PhD study want to complete inquiry-driven tasks with significant analytical depth and modeling rigor. Entrepreneurially-oriented students want to ideate, prototype, and pitch their solutions. The diversity of these pathways highlights a core instructional challenge as to how faculty might develop course-level tasks that remain feasible within a 15-week semester while addressing unique learning intentions and future paths.

The CDIO paradigm offers a principled basis for addressing this problem by centering engineering education around the Conceive–Design–Implement–Operate life cycle (Crawley et al., 2014). CDIO promotes contextually relevant, authentic, and practice-based learning and offers a systematic perspective on how engineering work should develop throughout the curriculum. Nonetheless, while there is robust commitment to CDIO worldwide, faculty often struggle with translating the paradigm into specific day-to-day course-level assignments. CDIO adoption largely reflects changes at the curriculum level, capstone-level and design-build, multi-year reforms, and there is a lack of day-to-day Praxis-level reform.

A review of CDIO literature shows that there are rules for integrated curricula and active learning (CDIO Standards 2, 3, 5, 7, 8, and 11) (CDIO Initiative, 2020), but far less so for how to structure assignments by type of course—foundational, system-oriented, practice-oriented, design-intensive, interdisciplinary, etc.—and how to balance such tasks under normal semester limits. As a result, assignments default to theoretical problem sets, isolated coding assignments, or descriptive reports that only partially capture CDIO (Felder & Brent, 2009).

This paper fills this gap by proposing a course-type based task design and orchestration framework that aligns with CDIO principles. Our contributions are:

1. A classification of nine types of undergraduate engineering courses that can be shaped by the feasibility of learning tasks.

2. A CDIO-aligned task taxonomy, including core tasks, communication tasks and reflexive tasks for each type of courses.

3. A task orchestration logic that prepares faculty to respond to different students' pathways, industry placement, research, and entrepreneurship within the semester boundaries.

4. Course level implementation evidence, from a Design of Machine Elements course, that shows how the framework can provide the facilitation of CDIO Design-Implement learning without curriculum changes.

BACKGROUND & RELATED WORK

The CDIO Initiative was founded to meet a commonly recognized problem: engineering graduates had a solid analytical base but were not prepared for engineering practice (Crawley et al., 2014). In CDIO, engineering curricula were reorganized around real-world professional processes using the Conceive–Design–Implement–Operate lifecycle and a set of standards that drive development of the curriculum, pedagogy, and assessment (CDIO Initiative, 2020). The most relevant of these standards at the course-level task design include: Standard 2 – Learning Outcomes: describing engineering practice knowledge, skills, and attitudes, Standard 3 – Integrated Curriculum: integrating courses, Standard 5 – Design–Implement Experiences: integrating engineering design into curriculum, Standards 7 & 8 – Integrated Learning & Active Learning: promoting learning by doing, Standard 11 – Assessment: promoting linkages between learning outcomes and assessment.

A large body of the current CDIO literature has focused on active learning, systems thinking, and real-life engineering experiences (Kraft et al., 2006), demonstrating the merits of problem-based and project-based learning as effective operationalizations of CDIO (Edström & Kolmos, 2014). Other literature addresses curriculum-level changes, multidisciplinary project courses, and multidisciplinary capstone design programs that include instruction on CDIO lifecycle tasks (Malmqvist et al., 2009). Previous conference research on CDIO has also considered course-level applications, especially with regard to design-build programs, integrated projects, and disciplinary course-level approaches (Bankel et al., 2005). But none of these studies generalize a model for the purpose of designing tasks at the course level.

Much of this CDIO literature takes a program or curriculum perspective, which can be resistant and time-consuming because it involves major structural or institutional changes. There has been less focus on a course-level operationalization of CDIO, the layer where most faculty work and where most student learning activities occur. Existing research documents success for task integration and active learning, but offers little guidance on how to select and interrelate tasks and activities in accordance with course type.

Simultaneous engineering education criticism notes that 1) traditional assignments overemphasize calculations and procedures, and 2) underrepresent real-world engineering problem solving (Felder & Brent, 2009). CDIO offers a way to integrate learning experiences, but in practice faculty have few structures to guide them in selection of tasks that integrate conceptual and applied learning (i.e. design, analysis, evaluation, communication).

This paper adds to CDIO literature by focusing the CDIO operationalization on the course level, rather than the curriculum level. Rather than adding to pedagogical theory, we integrate existing CDIO standards, active learning literature, and faculty practice in a practical framework to help instructors select and structure feasible and real tasks based on course type, workload, and desired alignment to CDIO lifecycle tasks.

FRAMEWORK DEVELOPMENT OVERVIEW

The process of framework development involved a systematic approach that consisted of four phases. First, there was a review of CDIO standards. Second, an institution-wide curriculum review and faculty interviews took place. Third, the engineering courses were classified according to type. Finally, the logic of task orchestration was synthesized. All phases are illustrated in Figure 1.

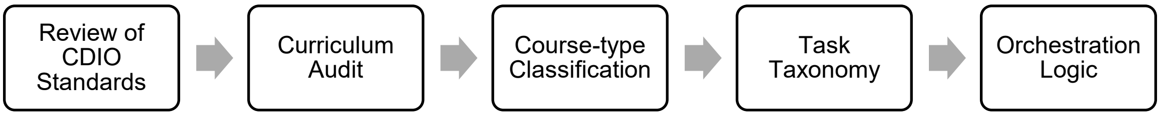

Figure 1. Framework development process

METHODOLOGY: DEVELOPING A COURSE-TYPE–DRIVEN TASK ORCHESTRATION FRAMEWORK

The development of the proposed task-orchestration framework was based on design-based research with the aim of developing a practical, course-level framework for using CDIO as an instructional model in undergraduate engineering programmes. The overarching goal was to develop a framework that allows faculty to design learning tasks that are realistic within semester-level time constraints, pedagogically coherent, and consistent with the Conceive–Design–Implement–Operate progression. The design phase included four interconnected steps: (1) a critical analysis of CDIO standards; (2) an institutional curriculum audit and faculty interviews; (3) categorization of types of engineering courses and associated task structures, and (4) synthesis of an orchestration logic that would facilitate the structured deployment of tasks within a variety of teaching contexts.

Review of CDIO Standards and Their Implications for Course-Level Learning Tasks

This step used a focused literature review of CDIO Standards and prior studies. Initial analysis involved a focused review of CDIO Standards, specifically identifying how the standards relate to learning task design. Standards 2 and 3 describe programme-level learning outcomes and integrated curricula. Standards 5, 7, 8, and 11 stress design–implementation experiences, integrated learning activities, active learning, and assessment alignment (CDIO Initiative, 2020). A central insight that emerged from this review was that CDIO has well-defined directional guidance on what students should experience. However, the standards provide relatively little direction on how designers should design learning tasks. For instance, the standards accept that rich and authentic learning tasks exist but do not provide guidance and direction on how learning tasks should be selected, structured, or sequenced with respect to credit load or learning objectives in a specific course. We used this insight as the conceptual basis for changing the focus of our enquiry from curriculum design to learning-task design as the operational definition of CDIO implementation.

Institutional Curriculum Audit and Faculty Consultation

This step used curriculum document analysis and structured faculty consultations. To investigate course-level implementation realities, an institutional curriculum audit of the undergraduate courses of the mechanical, electrical, electronics, computer science, and allied engineering programmes was carried out, assessing the course objectives, syllabi, assessment patterns, nature of existing assignments, and the constraints imposed by credit systems and semester duration. Structured consultations and workshops with the faculty members engaged in CDIO-aligned teaching work complemented this audit; these meetings uncovered the fact that instructors tend to use intuition rather than any common criteria when making assignment choices, and thus courses of the same type differed significantly from one another.

Faculty uniformly explained their challenge in cultivating active learning while covering theory, their uncertainty in distinguishing when to use mini-projects, case studies, simulations, or applied research, and the challenge in assessing design work in limited assessment time. These findings reinforced the need for a clear framework that would guide the faculty in making the right choice of task type based on the course type, depth of learning, and the relevant CDIO stage or stages.

Identification and Classification of Engineering Course Types

This step used iterative categorization based on recurring instructional patterns. The curriculum audit and faculty consultations clarified the need to categorize engineering courses according to recurring instructional types. Based on an iterative process, nine such course types were identified as prototypical of the undergraduate engineering experience: foundational-oriented, system-oriented, practice-oriented, design-oriented, advanced theory-oriented, interdisciplinary/application-oriented, skill/tool-oriented, societal/ethics/policy-oriented, and capstone/integrative. These types were not intended to be discipline-bound — they recur across programmes.

One important rationale was that not all courses can credibly support the same kinds of active learning. Demanding complete, fully-developed projects in foundational courses, or purely theoretical projects in practice-oriented courses, results in pedagogically infeasible instruction and a burden on students. The nine-course-type classification provides an essential structural foundation for task orchestration, by specifying what makes a task pedagogically viable in a particular course type.

Development of a CDIO-Aligned Task Taxonomy

This step used synthesis from active learning and CDIO literature. The next step was to develop a taxonomy of tasks that was consistent with CDIO principles and that could link course purpose and student activity. Using research on active learning (Felder & Brent, 2009; Edström & Kolmos, 2014), we organized tasks into three functional categories: core tasks, communication tasks, and reflective tasks. Core tasks are the central learning activity that students use to process engineering concepts or problems (study project, design projects, system analysis, research with applications). Communication tasks develop students' professional engineering communication skills (technical papers, conference presentations, posters, system demonstrations, conference papers where appropriate). Reflective tasks support metacognition by having students explain their reasoning, analyze trade-offs, and explore the learning process. The three-part structure creates assignments that are balanced and intentionally linked to CDIO Standards 7, 8, and 11, as well as ensuring that tasks can be completed within the semester.

Synthesis of Task Orchestration Logic

This step used integrative mapping of course type, CDIO stage, and learning depth. The last step in the methodology was the development of a task-orchestration logic for selecting and sequencing tasks within a course. Task orchestration is defined as the intentional combination of course type, target CDIO stage (s) within the CDIO life cycle, depth of learning (conceptual, analytical, design, implementation), time considerations and assessment feasibility. The orchestration logic obviates the need to prescribe common assignments across courses by allowing faculty to assemble combinations of tasks appropriate to the pedagogical intention of the course.

Foundational courses emphasize study and analytical tasks in the Conceive stage, design-oriented courses prioritize conceptual and detailed design tasks in Design, practice-oriented courses prioritize development and implementation tasks; and capstone courses address the whole life cycle. This logic supports constructive alignment of course objectives with CDIO intent and assessment without restructuring the curriculum, and can be implemented in the context of pre-existing academic pressures.

This orchestration approach is not rigidly defined, meaning while a course type has one dominant CDIO focus, an instructor can include additional secondary levels or alter task complexity.

Validation Through Course-Level Implementation

In order to demonstrate practical feasibility, the framework was implemented in actual courses, the most significant of which was a third-year Mechanical Engineering course titled Design of Machine Elements. The teaching staff used the orchestration logic to redesign assignments, while maintaining the existing syllabus and contact hours. The implementation demonstrated, as well as evidence that the framework is not only sound in theory but also operationally feasible and the basis for enhanced task design, clearer assessment, and improved CDIO alignment. The evidence and outcomes are reported in Section 7.

ANALYSIS AND RESULTS OF FRAMEWORK DEVELOPMENT

The analyses of the different developmental phases have led to the following outcomes: (1) the determination of nine kinds of engineering courses based on curriculum pattern analysis; (2) the identification of the orientation of key tasks that are relevant to each of the CDIO phases; (3) the creation of a three-level task taxonomy, comprising the core, communication, and reflective levels; and (4) the development of an orchestration logic.

THE TASK-ORCHESTRATION FRAMEWORK

We base the implementation of this idea on the fact that every undergraduate engineering course can be grouped into one of nine typical types which are identified in Table 1. For each of the nine course types, we define the dominant task orientation, the CDIO stages that the course naturally engenders and the typical workload for students for the course. This categorization is intended to help faculty to be clearer about what kind of engineering action this course is presumably best suited for priming and to guarantee that the CDIO stages Conceive, Design, Implement, Operate can be experienced in succession and not all at once in the same course. Table 1 presents the classification of engineering course types along with their defining characteristics, dominant task orientations, and associated CDIO stages.

Table 1. Course types and its task orientation

Course Type	Dominant Task Orientation	Primary CDIO Stage(s)	Typical Time Load
Foundational	Study and analytical tasks	Conceive	1–2 weeks
System-Oriented	System study and simulation	Conceive–Design	2–3 weeks
Practice-Oriented	Development and application	Design–Implement	3–4 weeks
Design-Oriented	Conceptual and detailed design	Design	3–4 weeks
Advanced Theory	Modeling and analysis	Conceive–Design	2–3 weeks
Interdisciplinary	Integrated projects	Conceive–Design–Implement	3–4 weeks
Skill/Tool-Oriented	Tool usage and workflows	Implement	1–2 weeks
Societal/Ethics	Case and proposal-based tasks	Conceive–Operate	2–3 weeks
Capstone	Full CDIO integration	Full lifecycle	Full semester

Each course type is defined as follows: Foundational: concept-building and analytical reasoning, System-oriented: system-level understanding and modelling, Practice-oriented: application and development tasks, Design-oriented: structured engineering design, Advanced theory: mathematical modelling and abstraction, Interdisciplinary: multi-domain integration, Skill/tool-oriented: tool proficiency and workflows, Societal/ethics: contextual and policy-driven analysis, Capstone: full lifecycle integration. Instructors can classify a course by identifying its primary learning objective (analysis, design, implementation, integration, or contextual evaluation) and aligning it with the closest dominant task orientation.

The second element of the framework is the task taxonomy, which organizes each course assignment into three mutually related elements, a core task, a communication task, and a reflective task. The purpose of this is to ensure that engineering action, engineering communication, and engineering judgment are part of the same assignment rather than being treated as discrete skill sets. The three components form a task unit, in which the core task is the center of engineering work, the communication task is the articulations of reasoning and technical clarity, and the reflective task is the meta-cognition and iterative improvement. Table 2 presents representative examples for selected course types; the framework supports extension to all nine types.

Table 2. Task Taxonomy (Representative Examples)

Course Type	Core Task Example	Communication Task	Reflective Task
Foundational	Study Project	Technical Report	Learning Reflection
System-Oriented	System Study Project	Architecture Presentation	System Reflection
Design-Oriented	Design Project	Design Review	Design Decision Reflection
Practice-Oriented	Development Project	Demo & Report	Implementation Reflection
Capstone	Integrated Project	Report, Defense & Poster	Capstone Reflection

The examples provided here are just some of the options. Several core tasks may be available for each course type, and communication and reflection aspects can also be tailored to fit the specific instructional setting.

The alignment of the course-type taxonomy and the task taxonomy results in a uniform and scalable approach to embedding CDIO into the curriculum. The key is that every course features the one integrated task unit and that the task unit always contains the same three elements, no matter the discipline. Indeed, what differs is the specifics of the core task: courses on foundational content need structured study and analytical problem framing; courses on the system need architectural reasoning; courses on practice need technical development; courses on design need structured engineering design; courses on advanced theory need modelling and interpretation; courses on interdisciplinary need integrative analysis; courses on society need contextual evaluation; courses on skills need the use of professional tools; and courses on capstone need full CDIO integration. Our framework addresses a persistent challenge in engineering education, namely that students are often bombarded by a disconnect assessment system where assignments are placed in no particular order in reference to CDIO and to each other. Now, with our framework, students see an orderly progression, where they learn to progress from conceptual analysis to system reasoning, from design exploration to design implementation and, finally, to professional reflective judgement. Further, our framework also addresses the day-to-day constraints of faculty: instead of requiring faculty to understand and design a meaningful assignment each semester, our framework provides a non-dimensional task structure for each course type. It does not require a faculty member to redesign the course or to convert each subject into some large project. Instead, the faculty member needs to assign the proper task unit for the course and this is automatically available from our framework. Thus, our framework is both rigorous and efficient and hence, scalable. Finally, our framework directly supports CDIO Standard 11 on assessment. Because every task unit contains explicit engineering, communication, and reflection elements, we can create rubrics that track consistency through analysis depth, engineering justification, communication quality, and the level of reflection maturity. We can create a clear trail for educational assessment.

COURSE-LEVEL CDIO IMPLEMENTATION AND RESULTS

In order to showcase the applicability of the proposed task-orchestration framework, it was implemented at course level in selected undergraduate engineering programs. The implementation aimed to investigate whether the framework could be embedded in a regular 15-week semester, replace traditional assignment format, and significantly operationalize CDIO concepts, without curriculum redesign. In this section, we describe the implementation context, the integration process and its instructional/learning outcomes, using the Design of Machine Elements course as a canonical implementation example.

Course Selection and Implementation Context

The choice of an appropriate course was essential to enable the demonstration that the curriculum framework is able to translate the CDIO model principles into specific, assessable learning tasks. The Design of Machine Elements (third year, fifth semester, undergraduate Mechanical Engineering Program) was chosen as this is a Design-Oriented course when applying the course-type classification that is proposed. Moreover, it was the obvious sequence for the CDIO Design stage.

The DME course requires students to integrate engineering mechanics, material behavior and failure theories for the design of mechanical elements and also to integrate knowledge from the prerequisite courses, including Matrices and Calculus, Python Programming, Engineering Mechanics and Strength of Materials. This integrative aspect made the course highly suitable to exemplify the use of a structure of tasks that activates and consolidates prior knowledge and simulates professional practice.

The chosen implementation context corresponded to a normal classroom and laboratory with no additional instructional hours or resources than what was already planned for the course. The existing assignments were restructured into one unit (CDIO-workflow) as required by the curriculum framework, but consisting of a single design task, a structured communication task and a reflective task. Hence, the implementation process represented a typical, scalable adoption process of the standards instead of a special, experimental offer.

Framework-Based Task Implementation

The task-orchestration framework was implemented in the course by redesigning an assessment strategy around a single cohesive design project, which was evaluated as a National Skills Qualifications Framework (NSQF) Level 6 design task, replacing an array of numerical, disconnected problem sets with a single, cohesive design project based on the CDIO Design and early Implement stages.

The design task required students to propose a mechanical robotic arm that could carry a given payload on defined operating conditions. For this, the students had to generate alternative design configurations, investigate material options, consider failure theories, and produce manufacturable specifications. They had to use engineering mechanics principles to compute the external loads on members and joints. They had to use strength of materials principles to compute internal stresses, deflections and factors of safety. They had to use Python programming to automate repeated calculations and investigate parameter sensitivity. The computational skills were tied to analytical thinking. They had to use CAD/CAE (Fusion 360) for modelling the assembly and to evaluate stress distribution using simulation.

Thus, the framework ensured that the technical activity is surrounded by a structure for communication and reflection. The students had to produce a comprehensive design report, in which they had to document assumptions, calculations, standards used, and the final design decisions. CAD models and simulation outputs were uploaded as evidence. The students had to give a formal design review presentation and to explain and defend engineering decisions. The students had to provide course design reflections, in which they had to evaluate trade-offs, constraints, and learning outcomes.

The new structure ensures that the course transfers from a typical topic-wise coverage towards a cohesive learning experience aligned to the CDIO domains. However, the content of the syllabus is not changed, nor have the instructional hours changed.

Implementation Outcomes and Observations

The course-level implementation was evaluated using qualitative analysis of student artefacts and structured student and instructor feedback.

Analysis of Student Artefacts

Student artefacts suggested a transition from procedural problem solving to integrated engineering reasoning. Design reports manifested the linkage of analytical calculations to design decisions, and explicit justification of assumptions and safety margins. CAD models and simulations showed proficiency in industry-standard tools and design validation approaches. Presentation artefacts indicated improved technical communication and communication of engineering reasoning to peers and faculty.

In general, the artefacts were the way a portfolio of professional engineering work would appear, rather than discrete academic exercises, suggesting successful activation of CDIO Design–Implement learning behaviors.

Faculty and Student Feedback

Faculty reported that the framework served as a clear and practical implementation structure at the course level. The separation between core, communication, and reflective tasks assisted in the design of their assignments and allowed clearer rubric development. Instructors also reported that the CDIO alignment became defensible and transparent for both instructional planning and accreditation documentation.

Students reported that the integrated project provided a clearer learning narrative for the semester. They mentioned that they better understood how their mathematical and mechanics concepts from earlier courses contributed to concrete design decisions. Students reported that the use of Python automation and CAD/CAE tools provided them with technical confidence and supported their confidence for preparation for industry placement and technical interview roles.

DISCUSSION

This section discusses the effectiveness of the proposed framework in addressing the aforementioned gap and its practical application to various instruction environments. The incorporation of the task-orchestration framework into a regular semester engineering course provides evidence that the framework can be applied in a scalable and pedagogically consistent manner. The results indicate that an organized, course-type–based task system can have substantive impact on student learning, faculty practice, and coherence between classroom activity and CDIO's experiential mission.

One of the central insights gained from implementation is the framework's ability to operationalize CDIO which starts as a philosophy at the curriculum level and translates into activity at the course level. While CDIO emphasizes integrated curriculum design and structured design–implementation experiences, instructors often have difficulty translating the standard into semester bounded assignments that fit within existing credit structures. By standardizing task structures that explicitly link course type to authentic and assessable activity, the framework offers instructors a pedagogical scaffold that is both extensible and easy to use. This minimizes dependence on faculty intuition and the "blank-page problem" that often limits CDIO aligned course design.

The framework furthermore contributes to instructional coherence by purposefully embedding the core, communication and reflective elements into every assigned task. In embedding a tripartite structure into instructional design, learners will not simply be engaging in engineering work but will be having opportunities to communicate processes and undertake reflective learning in relation to integrated learning experiences, active learning and robust assessment.

A significant advantage of the framework is in the management of prior learning. Design of Machine Elements implementation involved the activation of prerequisites in calculus, programming, mechanics, and strength of materials, which facilitated the vertical and horizontal integration that is in accordance with CDIO Standard 3.

When viewed from the faculty perspective, the framework reduced friction in designing aligned learning experiences. Instructors reported that having an unambiguous mapping between course type and appropriate tasks helped them design assignments and lowered the cognitive load of asking the instructor to generate rubrics. The alignment of tasks to CDIO stages also increased transparency of instructional planning and accreditation reporting. Incorporating reflection and communication within the task structure also facilitated a more comprehensive assessment of learning without exceeding existing workload expectations.

However, several limitations should be noted. First, although the implementation was proven feasible in the context of the Design of Machine Elements course, additional validation in other departments and across different course types (e.g., foundational, interdisciplinary, or societal) is discussed to assess scalability. Second, the current framework only implements the course level; a program- or year-long arrangement of task-based orchestration has yet to be addressed. Third, although artefacts showed good depth of student product and quality, longitudinal information will be required to investigate whether gains lead to competence development and impact performance in internship or capstone courses.

In the absence of these limitations however, this implementation provides strong evidence that the task-driven framework offers a feasible means to promote broader implementation of CDIO in engineering education programmes. The framework enhances course design and supports structured learning, under generally uncertain conditions, within the real-world context of semester-based teaching. The results suggest that the narrative supporting CDIO that it must necessarily be implemented as a large-scale curriculum reconfiguration is unnecessary; rather, it can be leveraged at the scale of an individual assignment, incrementally, course by course. With respect to implementation, the framework entailed little modification effort as instructors had to simply (1) classify the course, (2) choose an adequate task unit format, and (3) integrate assessment to the three task elements.

CONCLUSION

This article described a structured, course-level approach to operationalizing CDIO through a differentiated task-orchestration framework. By moving away from curriculum re-design to the orchestration of coherent, semester-slice learning tasks, the framework offers a useful tool for faculty to operationalize the CDIO lifecycle within existing undergraduate engineering courses. The classification of courses into nine instructional types paired with context-appropriate core, communication and reflective tasks supports constructive alignment, limits overload and ensures tasks are realistic for the engineering study context.

Its application in a conventional third-year Design of Machine Elements course showed that the framework was useful for orienting the integration of engineering analysis, design reasoning, and computational tools together with communication and reflective judgement, all within existing instructional parameters. Artefacts collected indicated measurable gains in coherence, depth and professional credibility, and faculty reported increased clarity in framing and reduced complexity in assignment design and assessment. The results, therefore, provide evidence that it is possible to align learning to CDIO by purposefully redesigning assignments, without requiring new semester hours or a change of curriculum.

For instructors looking to use the framework, the process requires three basic steps: determining the nature of the course, picking the appropriate task unit (core–communication–reflection), and matching it with the applicable CDIO phases. The modular design of the framework easily permits its application to other course types and engineering disciplines. Its straightforwardness and flexibility make it an immediately useful tool for supporting institutional change to CDIO, accreditation readiness and the long-term strengthening of the curriculum. Application of the framework will be broadened to more courses and programmes for future evaluation and will be linked to a comprehensive competency and progression model which is planned for journal publication.

ACKNOWLEDGEMENTS

The author(s) received no financial support for this work.

REFERENCES

Bankel, J., Berggren, K.-F., Engström, M., Wiklund, I., Crawley, E. F., Soderholm, D. H., & Östlund, S. (2005). Benchmarking engineering curricula with the CDIO syllabus. International Journal of Engineering Education, 21(1), 121–133.

CDIO Initiative. (2020). CDIO standards 3.0. CDIO Initiative.

Crawley, E. F., Malmqvist, J., Östlund, S., & Brodeur, D. R. (2014). Rethinking engineering education: The CDIO approach (2nd ed.). Springer.

Edström, K., & Kolmos, A. (2014). PBL and CDIO: Complementary models for engineering education development. European Journal of Engineering Education, 39(5), 539–555. https://doi.org/10.1080/03043797.2014.895703

Felder, R. M., & Brent, R. (2009). Active learning: An introduction. ASQ Higher Education Brief, 2(4), 1–5.

Malmqvist, J., Östlund, S., & Rosén, M. (2009). Implementing CDIO in engineering programs. International Journal of Engineering Education, 25(2), 1–12.

National Academy of Engineering. (2020). The engineer of 2030: Visions of engineering in the new century. NAE Press.

UNESCO. (2021). Engineering for sustainable development: Delivering on the Sustainable Development Goals (SDGs). UNESCO Publishing.

World Economic Forum. (2023). The future of jobs report 2023. World Economic Forum.

BIOGRAPHICAL INFORMATION

Vairavel Gurusamy is currently serving as Professor in Electronics and Communication Engineering Department and Director Academics and Educational Initiatives at SRM Institute of Science and Technology (University), Tiruchirappalli Campus, India. His research interests are Engineering Education, Instructional Technology, CDIO curricula design and implementations. He is a Senior IEEE member, Member-at-Large of CDIO International Council and a Fellow of Institution of Engineers.

Deiva Sundari P is the Principal of SRM Easwari Engineering College. She holds a Ph.D. in Power Electronics from the College of Engineering, Guindy, an M.E. in Power Electronics & Drives with a University Rank from Anna University, and a B.E. (EEE) Gold Medal from Madurai Kamaraj University. She has led several funded research and consultancy projects supported by agencies such as AICTE, DST, CVRDE, NIWE, and the Royal Academy of Engineering, UK.

Siva Kumar M is the Principal of SRM TRP Engineering College, Tiruchirappalli. He has over 30 years of experience spanning teaching, research, industry, and academic administration. He holds a B.E. in Mechanical Engineering from PSG College of Technology, an M.Tech. in Production Engineering with a University Gold Medal from National Engineering College, and a Ph.D. in Mechanical Engineering from Anna University. His research focuses on optimization, manufacturing systems, tolerance allocation, and meta-heuristic algorithms.

Corresponding author

Vairavel Gurusamy
Professor, Department of ECE,
Director Academics and Educational Initiatives, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, India 621105
vairavelg@ieee.org