

MAPPING THE CDIO CURRICULUM WITH NETWORK MODELS

Karen E. Willcox, Luwen Huang

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

ABSTRACT

This paper describes the creation of a mathematical model that represents the mapping of an undergraduate engineering degree program to the CDIO syllabus. A network model represents entities as nodes and relationships among entities as links between nodes. In the CDIO curriculum mapping network model, CDIO skills and program courses are modeled as entities. The network model defines directed relationships to represent the mapping between courses and CDIO skills. A relationship exists if a course addresses a CDIO skill. The result is a structured model on which we build scalable educational analytics and data visualization tools. The approach is demonstrated for the MIT undergraduate program in Aerospace Engineering.

KEYWORDS

Curriculum Mapping, Educational Analytics, Interactive Data Visualization, Standards: 1, 2, 3, 11, 12

INTRODUCTION

The CDIO (Conceive – Design – Implement – Operate) engineering education model provides a benchmark for assessing how a curriculum satisfies standards outlined in the CDIO syllabus (Bankel et al., 2005). The CDIO Initiative was conceived and developed in 2000 through a collaboration among the Massachusetts Institute of Technology (MIT), Chalmers University of Technology, Linköping University, and KTH Royal Institute of Technology, with financial support from the Wallenberg Foundation (Crawley et al., 2011). Since then, the CDIO framework has been adopted by engineering programs worldwide, playing a key role in curriculum design, teaching and learning, assessment and evaluation (Crawley et al., 2008).

The mapping of CDIO skills to courses within a curriculum is a core activity in the CDIO model that promotes discussion and analysis. Such a mapping is essential to support a gap analysis, to ensuring that skills are not applied before they are taught, and to balancing skill coverage across courses (Loyer et al., 2011, Crawley et al., 2014). To date, mapping is done by collecting tables of courses, program outcomes or other program attributes and corresponding CDIO skills (for examples, see Brennan et al., 2010, Campbell et al., 2009, Cloutier et al., 2010, Gunnarsson et al., 2007, Loyer et al., 2011). Table 1 is an example of such a table.

Table 1: Example of a typical mapping in table form of courses in a curriculum to CDIO skills

CDIO Skill	Course
2.1.1 Problem Identification and Formulation	Course A, Course B, Course F
2.2.2 Modeling	Course B

With only a table format, it is often difficult to conduct inference and pathway analyses. For example, a table does not easily reveal in what order skills are introduced, taught and applied throughout the program. It is also difficult to gain a holistic sense of a program's topology with respect to CDIO coverage, answering questions such as: what is the distribution of skill coverage across courses? Because CDIO skills are cross-curricular and crosscut disciplines (Crawley et al., 2014), it is potentially useful to visualize an entire program's mapping to CDIO skills, beyond the table-based views of traditional spreadsheets.

In this paper, we introduce a mathematical network model that represents the mapping of a program to the CDIO syllabus. We apply our model to the MIT Aerospace Engineering program, mapping courses in the program to the CDIO syllabus. We show that the resulting network-based structure supports scalable visualization and analytics.

CDIO CURRICULUM MAPPING MODEL

In this section we formulate a network model for representing an educational program and its mapping to the CDIO syllabus. The two key elements of a network model are *nodes* and *edges*. A network structure represents entities as nodes and relationships between entities as edges between nodes.

In our CDIO curriculum mapping network model, we define four different types of entities:

- *content*: entities of type *content* represent a curricular unit of interest. This is often a course in curriculum design and accreditation processes; however, other curricular entities can be chosen as well, for example, learning units, resources, modules, etc.
- *CDIO skill*: these entities represent the elements of the CDIO syllabus. Here we use CDIO Syllabus v2.0.
- *group*: these entities represent units of organization in the curriculum. For example, courses may be grouped according to the type of course (elective, core, capstone, etc.). We may wish also to group the CDIO skills in clusters.
- *program*: these entities represent the programs being modeled.

Each entity is represented as a node in the network model. Information on the entity is modeled as a property of the node. The type of entity is one piece of information. For courses, we may include other properties such as the website URL of the course, the number of credits the course is worth, etc.

In our CDIO curriculum mapping network model, we define two different types of relationships:

- *addresses*: this is a directed relationship used to indicate that a course addresses a CDIO skill. If Content A *addresses* CDIO skill 2.1.1, we create a directed edge between the node of Content A and the node of CDIO skill 2.1.1 (Figure 1). The *addresses* edges are assigned a weighting to indicate how strongly a course addresses an outcome.



Figure 1: Content A addresses Skill 2.1.1

- *has-parent-of*: this is a directed relationship that specifies organizational hierarchy. We use these relationships to relate courses to a group, and groups to a program. We create a directed edge of *has-parent-of* type from Content A to Group X, if Content A belongs to Group X (Figure 2).

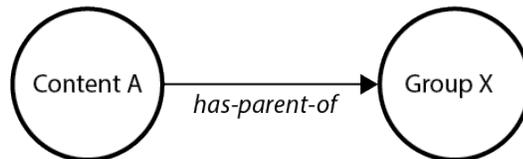


Figure 2: Content A has the parent of Group X, i.e., it belongs in Group X

Modeling the program in this way gives us a structured foundation for visualization and analytics. In particular, we can now apply to our network model the powerful tools of graph visualization and graph analytics to reveal insight into program structure and to analyze how the program maps to the CDIO syllabus. The next section shows this in detail for a case study of an undergraduate aerospace engineering degree program.

MAPPING A CDIO-BASED AEROSPACE ENGINEERING PROGRAM

In this section we model the MIT undergraduate degree program in Aerospace Engineering and its mapping to the CDIO syllabus. We demonstrate visualization and analytics using the network model.

In this example, we model the degree program at the level of individual courses – our content nodes are undergraduate courses taught in the Department of Aeronautics and Astronautics at MIT¹. The first step in creating the network model is to identify all the entities. In the model we have 20 courses, 87 CDIO skills, 17 groups, and 1 program. Each one of these entities is a node in the network model.

The second step in creating the network model is to identify all the relationships. In this case we identify 124 *has-parent-of* relationships that specify organizational groupings. We define a parent group for each course in the categories “Aerospace Engineering Core,” “Aerospace Engineering Capstone” or “Aerospace Engineering Elective.” Similarly, we use *has-parent-of* relationships to group the CDIO skills into their 14 categories (“2.1 Analytical Reasoning and Problem Solving” etc.). Each of the 17 groups also has a *has-parent-of* relationship that links it to the overall program.

We then define 516 *addresses* relationships that map each individual course to the set of CDIO skills. These mappings were created using data collected from faculty members during the end-of-semester course reflective memo process. Each faculty member is asked to specify which of the CDIO skills they introduce, teach and apply in their course. Table 2 shows a sample of the raw data collected from the faculty member teaching course 16.90 Computational Methods in Aerospace Engineering.

¹ Note that the Aerospace Engineering curriculum includes courses taught by other departments (mathematics, physics, chemistry, biology, computer science, humanities, arts and social sciences). Those courses are not included in the CDIO mapping presented here.

Table 3 summarizes the features of the overall mapped data set, including the attributes that are assigned to each entity and relationship. The network model has in total 125 entities and 640 relationships.

Table 2: A sample of the raw data mapping course 16.90 Computational Methods in Aerospace Engineering to the CDIO skills.

2.1 Analytical Reasoning and Problem Solving	I	T	A
2.1.1 Problem Identification and Formulation		X	
2.1.2 Modeling		X	
2.1.3 Estimation and Qualitative Analysis		X	
2.1.4 Analysis with Uncertainty		X	
2.1.5 Solution and Recommendation		X	
2.2 Experimentation, Investigation, and Knowledge Discovery	I	T	A
2.2.1 Hypothesis Formulation			
2.2.2 Survey of Print and Electronic Literature			
2.2.3 Experimental Inquiry			
2.2.4 Hypothesis Test and Defense			

Table 3: Summary of mapped MIT aerospace engineering CDIO curriculum

Entities	Count	Attributes
Content	20	URL, Keywords
CDIO skill	87	
Group	17	
Program	1	
Relationships	Count	Attributes
<i>has-parent-of</i>	124	
<i>addresses</i>	516	Introduce/Teach/Apply category

Figure 3 is a visualization of the mapped dataset. CDIO skills are visualized as small red nodes, grouped into their categories. Courses are visualized as larger nodes, grouped as Core, Capstone and Elective. The visualization is interactive and accessible online at mapping.mit.edu/cdio-mapping. To create this visualization, the authors used the online Rhumbl visualization tool²; the spreadsheet used to create the visualization is also available at mapping.mit.edu/cdio-mapping for download, and can be used as a template for other researchers who wish to create visualizations of their own CDIO data.

² Rhumbl (rhumbl.com) is a free online visualization tool that lets users create interactive network visualizations from spreadsheets. It was created by the authors in 2015.

Figure 4 shows a snapshot of the interactivity displaying the CDIO skills that are addressed by the selected course (here 16.06 Principles of Automatic Control). Figure 5 shows a snapshot of the selected CDIO skill of The Design Process Phasing and Approaches (4.4.2) and the courses that address it: Unified Engineering Fluid Dynamics (16.003), Unified Engineering Thermodynamics (16.004), Structural Mechanics (16.20), Real-Time Systems & Software (16.35), Autonomy (16.410), Experimental Projects (16.62x), Flight Vehicle Development / Engineering (16.82x) and Space Systems Development / Engineering (16.83x).

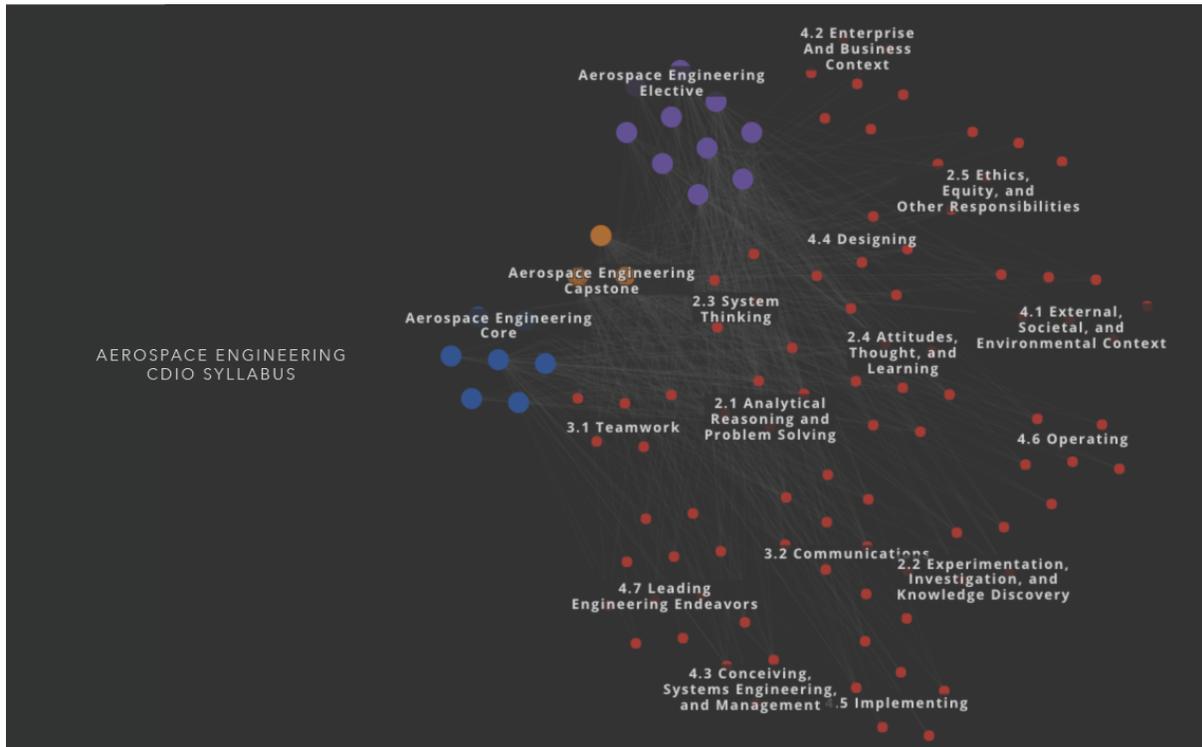


Figure 3: Visualization of the mapping of MIT aerospace engineering courses to the CDIO syllabus.

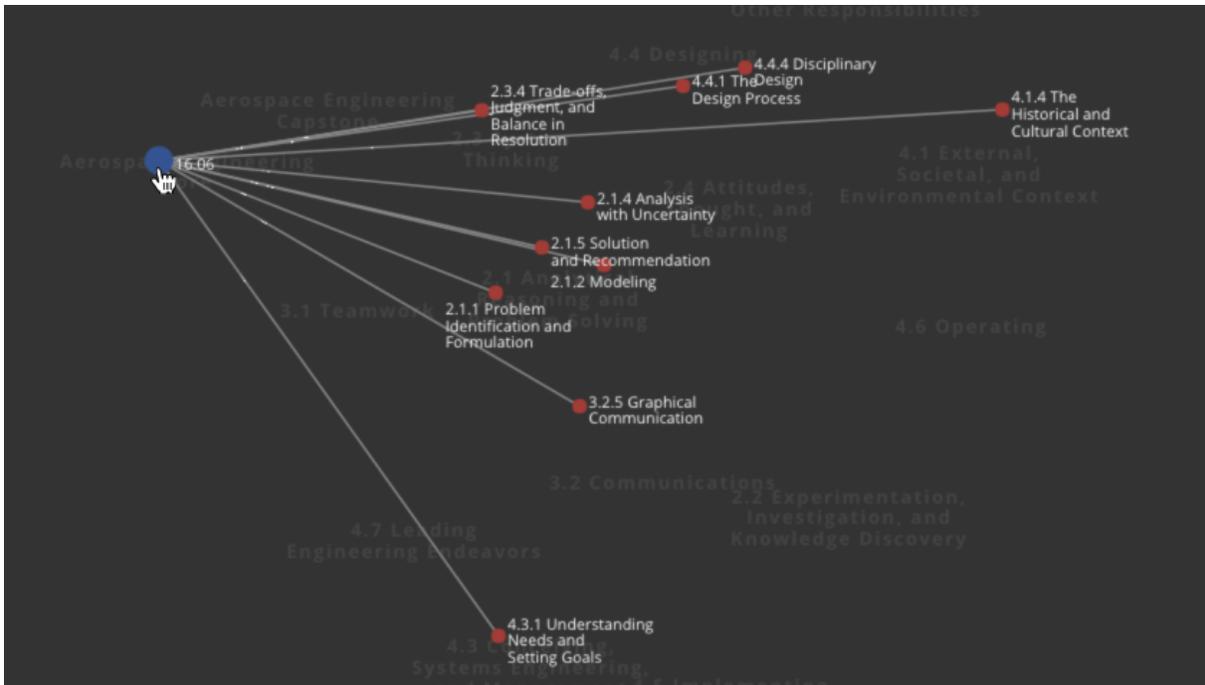


Figure 4: Mouse-over of course 16.06 (Principles of Automatic Control) in the interactive visualization shows the CDIO skills addressed by the course.

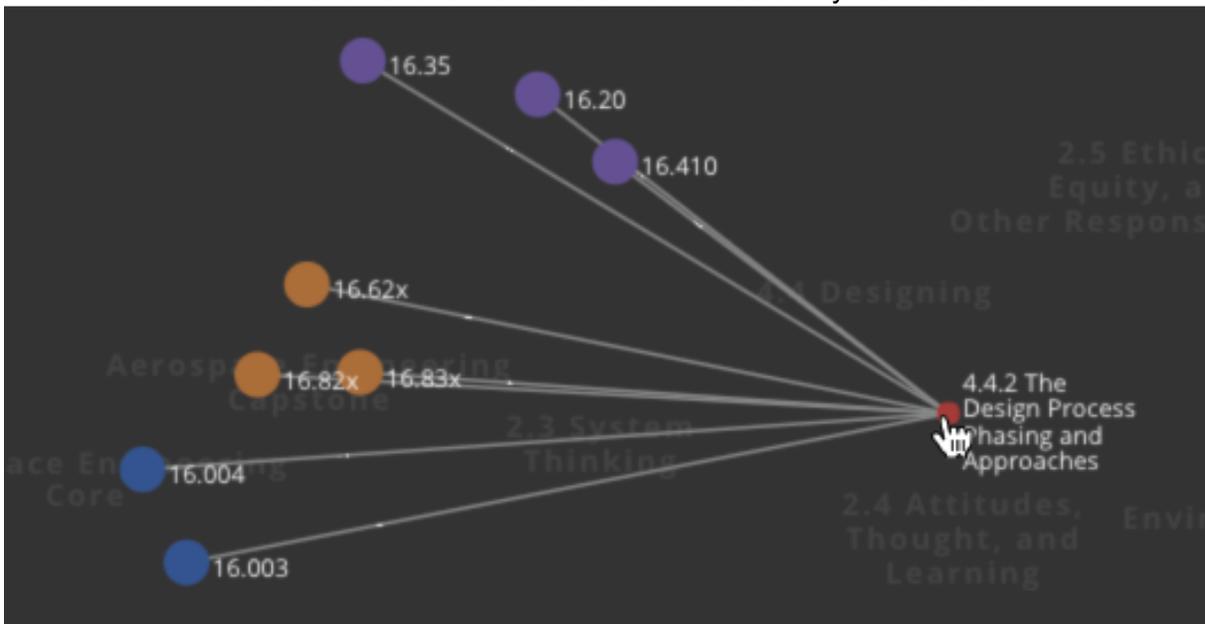


Figure 5: Mouse-over on CDIO skill 4.4.2 in the interactive visualization shows the courses that address the skill.

This network model provides a basis on which to conduct analysis of the program and its mapping to the CDIO syllabus. For each CDIO skill we can analyze its coverage in the program. The indegree of a node in the network model is a count of the number of incoming edges, while the outdegree of a node is a count of the number of outgoing edges. Thus, the indegree of a CDIO skill node specifies the number of courses that address that skill.

Similarly, the outdegree for *addresses* relationships of a course node specifies the number of CDIO skills addressed by that course. In both cases, we could also use the attribute of Introduce/Teach/Apply to break down these counts into their ITA categories.

Figure 6, Figure 7 and Figure 8 show the indegree calculations for the MIT aerospace engineering data set. As expected, the courses map strongly to the skills of Analytical Reasoning and Problem Solving (2.1), as well as skill Critical Thinking (2.4.4). The program courses also have a strong coverage of Teamwork (3.1), Communications (3.2), Conceiving, Systems Engineering, and Management (4.3), and Designing (4.4). We note that while the plots show some CDIO skills with no coverage in the mapped courses, MIT students likely receive exposure to these other skills in their other required courses outside the department.

In general, this kind of analysis will reveal areas of strength and potential gaps of coverage in a curriculum. This analysis is also useful in considering the effects of curriculum redesign on coverage of the CDIO syllabus. Noting that students select only a subset of courses, these scores could also be combined with student enrollment data to analyze student exposure to the CDIO syllabus, either for an individual student or in aggregate across the student population.

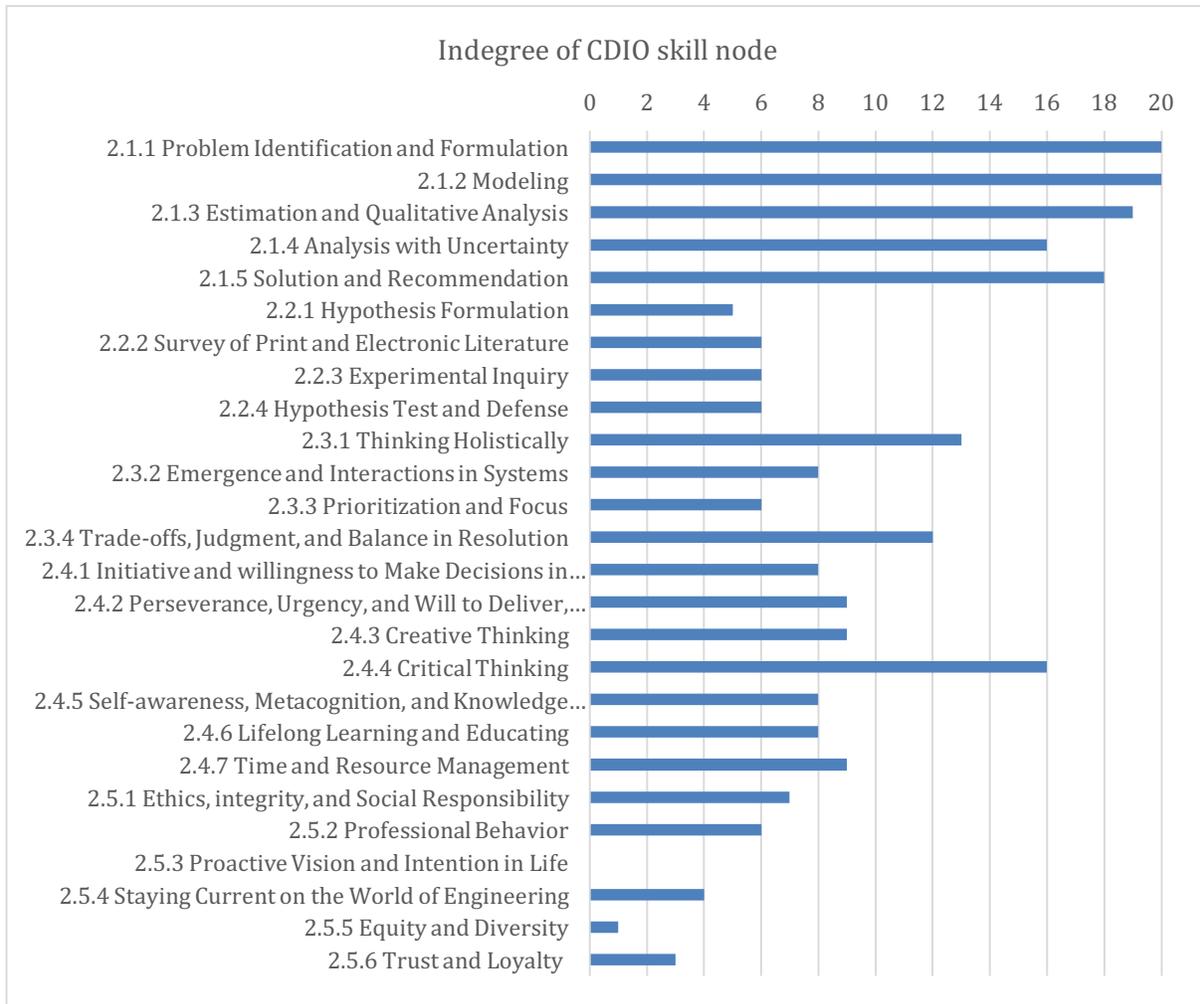


Figure 6: The indegree of CDIO skill nodes (here categories 2.1 through 2.5) specifies the number of aerospace engineering courses that address each skill.

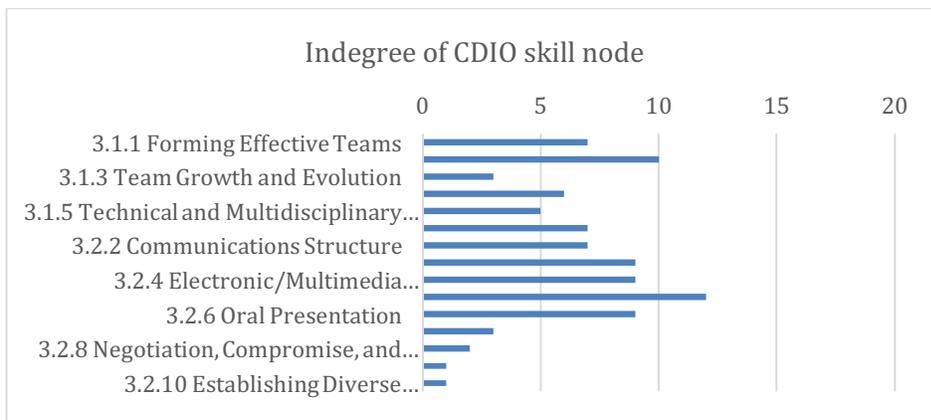


Figure 7: The indegree of CDIO skill nodes (here categories 3.1 and 3.2) specifies the number of aerospace engineering courses that address each skill.

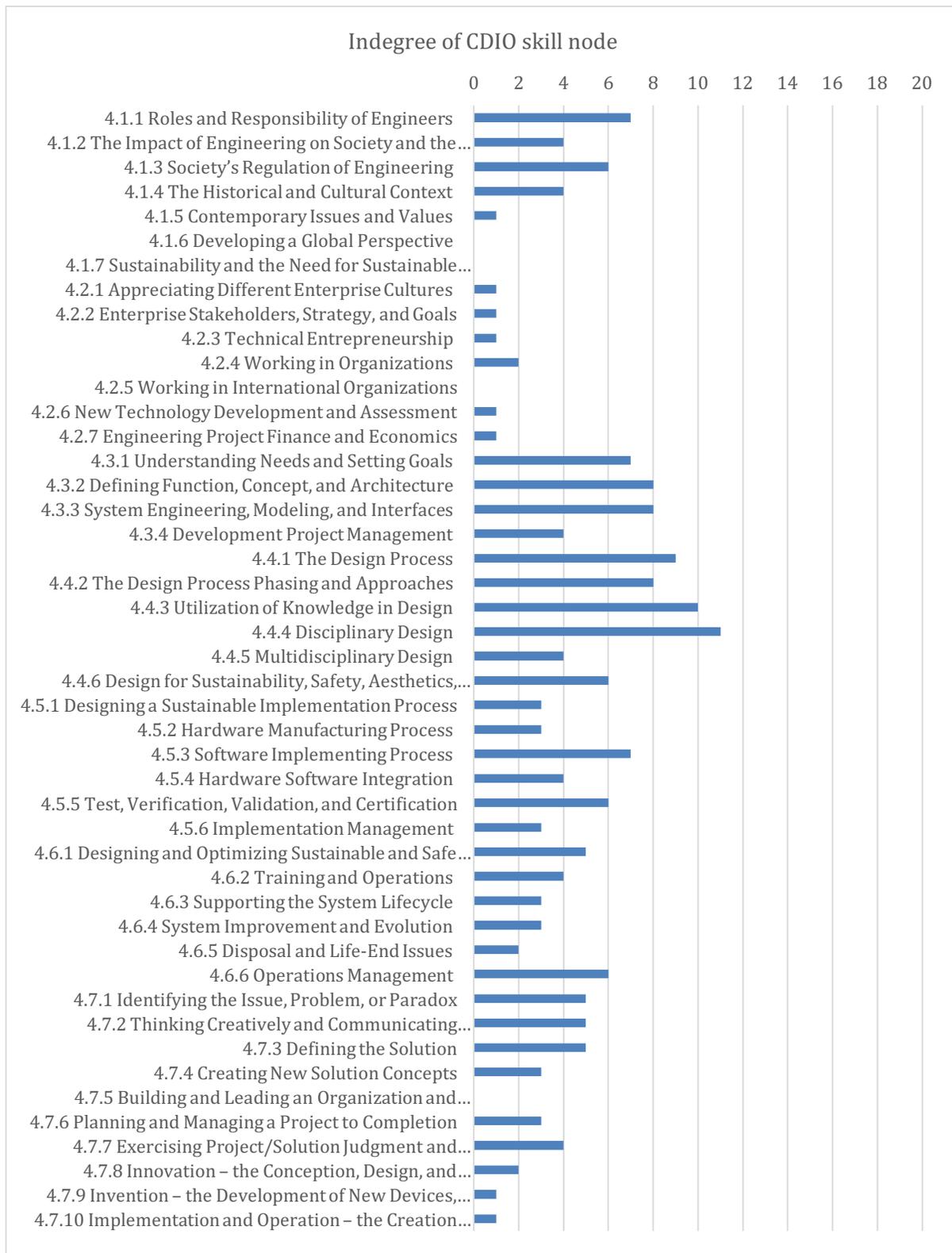


Figure 8: The indegree of CDIO skill nodes (here categories 4.1 through 4.7) specifies the number of aerospace engineering courses that address each skill.

An outdegree analysis of the courses shows that the capstone courses address more CDIO skills than any of the other courses in the program. This result is consistent with the design of the aerospace engineering curriculum, which strongly embeds CDIO principles within these capstone courses. We do not present the detailed quantitative results for the course outdegree analysis because of the possible noise introduced by having each course mapping done by a different faculty member. The data support analysis of general trends, but a detailed numerical comparison between courses is not meaningful with this particular data set.

CONCLUSION

A network model provides a basis on which to analyze and visualize the mapping of an educational program to the CDIO syllabus. In the example presented here, the curriculum was modeled at the level of a course, but the network model has the flexibility to model at any level specified by the modeler. In performing a detailed curriculum gap analysis, it may be desirable to model at a more granular level, such as learning units, modules, course learning outcomes, etc. This paper showed how analysis of node degree provides insight into the relationship between program courses and CDIO skills. With the network model in hand, other graph analysis techniques can also be employed. Path analyses may be of specific interest, particularly if the curriculum model is combined with student data. The visualization can play an important role in sharing the CDIO mapping among instructors and program coordinators (and indeed among different institutions as well). Since often the mapping for each course is done by an individual instructor, interpretations can vary and thus impact the quality of the resulting integrated dataset. Having the data available in a clear, interactive and searchable visualization can greatly facilitate the conversations needed to address this issue.

REFERENCES

- Bankel, J., Berggren, K., Engström, M., Wiklund, I., Crawley, E., Soderholm, D., Gaidi, K. & Östlund, S. (2005). *Benchmarking Engineering Curricula with the CDIO Syllabus. The International Journal of Engineering Education, Vol. 21 No. 1.*
- Brodeur, D., Crawley, E., Ingemarsson, I., Malmqvist, J., Östlund, S.. (2002). Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition.
- Brennan, R. & Hugo, R. (2010). *The CDIO Syllabus and Outcomes-Based Assessment: A Case Study of a Canadian Mechanical Engineering Program. Proceedings of the 6th International CDIO Conference, École Polytechnique, Montréal, June 15-18, 2010.*
- Campbell, D., Dawes, L., Beck, H., Wallace, S., Boman, M. & Reidsema, C. (2009). *Graduate Attribute Mapping With the Extended CDIO Framework. 20th Australasian Association for Engineering Education Conference, University of Adelaide, 6-9 December 2009.*
- Cloutier, G., Hugo, R., Sellens, R. (2010). *Mapping the Relationship between the CDIO Syllabus and the 2008 CEAB Graduate Attributes. Proceedings of the 6th International CDIO Conference, École Polytechnique, Montréal, June 15-18, 2010.*
- Crawley, E., Malmqvist, J., Östlund, S., Brodeur, D. & Edström, K. (2014). *Rethinking Engineering Education, Second Edition.* Switzerland: Springer.

Crawley, E., Malmqvist, J., Lucas, W. & Brodeur, D. (2011). *The CDIO Syllabus v2.0. An Updated Statement of Goals for Engineering Education. Proceedings of 7th International CDIO Conference, Technical University of Denmark, Copenhagen, June 20 – 23, 2011.*

Crawley, E., Brodeur, D. & Soderholm, D. (2008). The Education of Future Aeronautical Engineers: Conceiving, Designing, Implementing and Operating. *Journal of Science Education and Technology*, Volume 17, Issue 2, pp 138–151, April 2008.

Gunnarsson et al. (2007). *Large Scale Use of the CDIO Syllabus in Formulation of Program and Course Goals, Proceedings of the 3rd International CDIO Conference, Massachusetts Institute of Technology, Cambridge MA, June 11-14, 2007.*

Loyer, S., Muñoz, M., Cardenas, C., Martinez, C., Cepeda, M., Faundez, V. (2011). *A CDIO Approach to Curriculum Design of Five Engineering Programs at UCSC. Proceedings of the 7th International CDIO Conference, Technical University of Denmark, Copenhagen, June 20–23, 2011.*

BIOGRAPHICAL INFORMATION

Karen E. Willcox, Ph. D. is Professor of Aeronautics and Astronautics and Co-director of the Center for Computational Engineering at the Massachusetts Institute of Technology. Her research program develops computational methods to support decision-making in aerospace system design and operation. She also leads educational research through the MIT Mapping Lab, including a Department of Education First in the World project. She is currently co-Chair of the MIT Online Education Policy Initiative and Chair of the MIT OpenCourseWare Faculty Advisory Board. In 2013-2014, she served as co-Chair of the Institute wide Task Force on the Future of MIT Education.

Luwen Huang is a specialist in interactive data visualization, software engineering, and educational technology design and development. She is the technical lead of the MIT Mapping Lab, which develops technologies for educational modeling, data visualization and analytics.

Corresponding author

Professor Karen E. Willcox
Massachusetts Institute of Technology
77 Massachusetts Avenue 37-447
Cambridge, Massachusetts, USA 02139
1-617-253-3503
kwillcox@mit.edu



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/).