An investigation on integration of computational thinking into engineering curriculum at delft university of technology

Zhang, X.; Valle Torre, M.; Specht, M.M.

DOI
10.5821/conference-9788412322262.1230

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 50th Annual Conference Universitat Politècnica de Catalunya · BarcelonaTech (UPC)

Citation (APA)
Zhang, X., Valle Torre, M., & Specht, M. M. (2022). An investigation on integration of computational thinking into engineering curriculum at delft university of technology. In H.-M. Järvinen, S. Silvestre, A. Llorens, & B. Nagy (Eds.), *Proceedings of the 50th Annual Conference Universitat Politècnica de Catalunya · BarcelonaTech (UPC)* (pp. 890-901) https://doi.org/10.5821/conference-9788412322262.1230

SEFI 2022

SEFI ANNUAL CONFERENCE
19-22 September 2022
BARCELONA

50th Annual Conference of The European Society for Engineering Education

19-22 September, Barcelona, Spain

Towards a new future in engineering education, new scenarios that European alliances of tech universities open up

PROCEEDINGS

50th Annual Conference of The European Society for Engineering Education

19-22 September, Barcelona, Spain

Towards a new future in engineering education, new scenarios that European alliances of tech universities open up

SEFI 50th Annual Conference
Universitat Politècnica de Catalunya · BarcelonaTech (UPC)
19 –22 September 2022

ISBN: 978-84-123222-6-2

Editors:
Hannu-Matti Järvinen, Santiago Silvestre, Ariadna Llorens and Balàzs Nagy

Managing editor:
José Miguel Quiñones

Technical editor:
Ernests Edvards Zalitis

Printed by: Artes Gráficas Torres S.L., Huelva 9, 08940 Cornellà de Llobregat, Spain

The manuscript was closed on 30 November 2022.

AN INVESTIGATION ON INTEGRATION OF COMPUTATIONAL THINKING INTO ENGINEERING CURRICULUM AT DELFT UNIVERSITY OF TECHNOLOGY

X. Zhang [1]
Delft University of Technology
Delft, The Netherlands
https://orcid.org/0000-0003-0951-0771

M. Valle Torre
Delft University of Technology
Delft, The Netherlands
https://orcid.org/0000-0002-0456-8360

M. Specht
Delft University of Technology
Delft, The Netherlands
https://orcid.org/0000-0002-6086-8480

Conference Key Areas: *Fostering Engineering Education Research & Curriculum Development, Engineering Skills, Lifelong Learning*
Keywords: *Engineering Education; Computational Thinking; Curriculum; Case Study*

ABSTRACT

Our life is surrounded by digital devices. Engineering education is one of the cornerstones in higher education for future generations and computational thinking (CT) is deemed as a core component in various engineering curricula. The Delft University of Technology (TU Delft), is the largest technical university in the Netherlands and computing; computational concepts and activities have been integrated into curriculum for years at TU Delft. However, there is not a comprehensive investigation on integration of CT into Engineering Curriculum, this paper presents a case study of Master's level engineering curricula investigating: 1) to what extend CT components are integrated; 2) in what way CT is interpreted and integrated in the curriculum; 3) what educational and assessment methods have been used. The results show that CT has been largely integrated into the investigated curriculum mostly with lectures being the educational method and programming assignments as a method for the assessment. Our analysis shows that understanding the context and patterns in problems and solutions was important in different courses and engineering disciplines, indicating possible directions for integration of CT into curriculum.

1 Corresponding Author
X.Zhang
X.Zhang-14@tudelft.nl

doi:10.5821/conference-9788412322262.1230

1 INTRODUCTION

The nature and shape of engineering education is an ongoing topic of debate among engineering faculty members, professionals and practising engineers [1]. It appears to be that the enormous changes in societal dynamics and the development of technology also lead to transformation of the engineering curricula. In both the United States and Europe, an emphasis has been placed on digital skills and literacy. In 2016, Barack Obama launched the 'Computer Science for All' initiative. The aims of this initiative were to allow American students from kindergarten through high school to learn computer science and acquire computational thinking (CT) skills. Computer science was referred to as a new basic skill necessary for economic opportunity and social mobility. As for the European Union, the Joint Report of the Council and the Commission on the Implementation of the Strategic Framework for European Cooperation in Education and Training (ET2020) highlights the importance of digital competences. In addition, the study Developing Computational Thinking in Compulsory Education by the Joint Research Centre, which is the European Commission's science and knowledge service, suggests that CT is a subject of importance within the European Union.

Addressing this emphasis on digital competences and computational thinking, in the Netherlands 4TU has been created to foster the collaboration between the four universities of technology (TU Delft, Eindhoven University of Technology, University of Twente and University of Wageningen). In the vision for Higher Engineering Education written by Kamp (2016), the 4TU Centre of Engineering Education in the Netherlands is mentioned as a "Free-Spirits" think tank which aims to develop suitable higher education scenarios by 2030 by examining the appearance of new engineering profiles in the coming 10 to 15 years. Inspired by the Greek Philosopher Heraclitus stating "the only constant in life is change", its vision states that digitalization is one of the driving forces which makes our world become more uncertain, complex and ambiguous. This does not only impact how people live, but also requires the future generations of students to be empowered with lifelong-learning and general problem-solving competencies including the use of digital and computational tools.

CT was first mentioned by Papert (1980) and then advocated by researchers and practitioners since Wing claimed it as "a must-have skill for everyone" living in the 21st Century rather than solely for computer scientists. Ever since, there has been extensive debate and research on the definition of CT and its relevance for education. Several definitions are used in the field, for example, Wing (2011) defined CT as a set of problem solving skills with which the formulation of solution for problems can be carried out by computing agents (either human or mechanics computing machines). Unlike Wing (2011)'s definition which is descriptive and theoretical, some operational definitions attempt to identify more granular constructs that are more practical. Brennan and Resnick (2012)'s three-dimensional framework consists of CT concepts, CT practices and CT perspectives as well as the four compositional frameworks with problem decomposition, pattern recognition, abstraction, algorithm defined by BBC(2018) are frequently used in research and practices. While efforts on concretizing the definition of CT are still ongoing, the importance of helping people adapt and prepare future generations to live in digital society is widely recognized. As a result, researchers, educators, policy makers and practitioners from both Science, Technology, Engineering and Mathematics (STEM) and non-STEM backgrounds are currently investigating the integration of CT into different fields.

Bearing in mind that fostering students' CT skills is a strategic focus of the 4TU, in this work we aim to investigate their integration in engineering curricula. Considering the effort needed for investigation of all relevant faculties, we limit our case study to the MSc programmes of the two [E1] faculties of TU Delft that incline to integrate CT into their curriculum, and we specifically aim to answer the following research questions:

RQ1. To what extent is CT reflected in the curricula of the TU Delft?
RQ2. In what way does the interpretation or reflection of CT differ per faculty?
RQ3. Which educational and assessment methods are used in CT-integrated courses?

2 METHODOLOGY

2.1 Research Scope and Source Data

The scope of this study was limited to the MSc programmes of two TU Delft faculties: Architecture and the Built Environment (ABE), and Electrical Engineering, Mathematics & Computer Science (EEMCS). As our data source we used the TU Delft study guide, in which we initially performed a keyword-based search and then analysed the course data towards answering the research questions. The search criteria were informed after consultations with faculty members. The TU Delft study guide[2] is the formal collection of all courses and study programmes offered by the different faculties of TU Delft. Each course is listed and described in this database with information such as its overview on course design and general course arrangement. A search functionality offers the opportunity to search the course descriptions per faculty for one or multiple keywords. Hence, the study guide was used as the basis for this systematic research into how computational thinking is reflected across the faculties of the TU Delft.

2.2 Identification of Relevant Keywords

To identify courses that discussed CT explicitly, the keywords "computational thinking", "digital skills", and "digital competency" were applied. Besides identifying courses that explicitly discussed CT, we were interested in identifying the courses that implicitly included CT. To identify those courses, we leveraged the operationalization of CT devised by BBC (2018), which comprises problem decomposition, pattern recognition, abstraction and algorithm design. For each of the four components of CT, keywords were defined that signalled the presence of that step of CT (Table 1 in the Appendix). The asterisk (*) in some keywords ensures courses are found where both words occur in the text, although not

[2] https://studiegids.tudelft.nl/bbDefault.do?SIS_SwitchLang=en

sequentially. In addition, it allows for finding both UK and US spelling in e.g. mode*ing. In addition to these general keywords, after consultation with an associate professor for the faculty of Architecture and the Built Environment, the keywords “parametric”, “computational”, “ algorithm”, and “simulation” were added as ABE-specific keywords.

2.3 Filtering of the Relevant Data

After performing the keyword search, several filtering steps were applied (Figure 1 in the Appendix). First, BSc courses were discarded since we were solely interested in MSc courses. Since a considerable number of courses within EEMCS that included one of the CT components, only the courses with at least two CT components were evaluated. Then, while the presence of one or more CT components under each component in the course description signalled the possibility of CT, it was found that merely the appearance of those components were not sufficiently specific to signal the presence of CT. Therefore, the keywords and its corresponding CT components were applied, and an evaluation of the course description in study guide was performed to identify the courses reflecting CT. The courses were scored on the presence or absence for each of the four components of CT. The course descriptions which at least signalled the presence of two out of the four components of CT were included in our validation set while courses with less than two components were discarded. The validation sets were sent to three CT experts (second evaluators) for evaluation together with the operationalization of CT. The courses that were rated by at least three out of four raters as CT were included in the final selection.

2.4 Data Extraction and Synthesis

The selected courses were analysed with the information extracted from CT courses. Types and categories of the assessment method, educational methods, courses and other categorizations were based on observation from the dataset and aggregated by one of the authors with consultation from experts when necessary. Table 2 in Appendix provides an overview of the extracted information per CT course (with the last row presenting the information aggregated based on descriptions of the courses).

3 RESULTS

3.1 Inter-rater Reliability on the Identification of the Relevant Course

A low Fleiss’ kappa inter-rater reliabilities of 0.33 and 0.17 for the courses of the faculties ABE and EEMCS, respectively. These low values indicate that, although the operationalization of CT of this study was provided to the raters, their interpretations of CT were still divergent. Apparently, the provided operationalization left room for interpretation, which led each rater to use its specific background knowledge on CT. Zooming in on the ratings, we observe e.g. that one expert utilised a broad perspective and considered almost all courses of the test sets to be CT, while another expert utilised a narrow perspective and only scored about half of these courses to be CT. Clearly, their interpretations of CT differ greatly.

3.2 CT in the Curricula

A total of 15 ABE MSc courses under two MSc programmes (Geomatics and Architecture, Urbanism and Building Sciences (AU&B)) were designated as CT courses while a total of 27 EEMCS MSc courses were designated as CT courses (an additional filtering step was applied compared to ABE). The keywords that were found in the course descriptions of the CT courses are visualised in Figure 2a and Figure 2b (can be found in Appendix), with the font size correlating to frequency. Compared to the keywords from ABE, we can see "algorithms" and "algorithm design" were much more frequent for EEMCS. Meanwhile, Of the three MSc programmes analysed, unsurprisingly, Computer Science contained the highest number of CT courses. Due to the set-up of the EEMCS MSc programmes, all courses were compulsory choices courses.

Figures 3 and 4 demonstrate the different types of CT courses within ABE and EEMCS respectively. Most courses at ABE were about learning to use a computer program or a modelling task that contributes to analysing the environment or the design of new buildings. For example, in Digital Terrain Modelling students were taught to use datasets to reconstruct a terrain and use these for applications related to the built environment, and in Geomatics as support for energy applications students are taught to build 3D city models. Another type of CT courses at ABE were courses that comprised large projects with a computational component of varying extent. For example, in MEGA[3] students designed a special big building in multi-disciplinary teams. Within this team, just one of the students was responsible for the computational design. On the other hand, in Earthy the design of the building is completely computational. Finally, the course Operations research methods was specifically on teaching research methods and using mathematical modelling to make decisions.

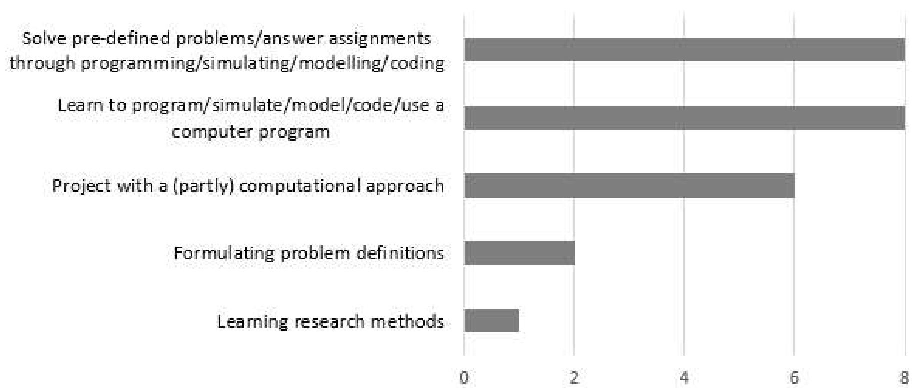

Fig. 3. ABE – Types of reflection of CT in courses

As for EEMCS, the largest part of the courses includes learning about the context and patterns in problems and solutions; frequently in the form of students being taught to recognise which algorithms to use for which problems. For example, in Object Classification

[3] MEGA is a collaborative integral multi-disciplinary design of a special big and/or tall building.

with Radar two of the four learning goals specifically refer to recognizing patterns in the data and the type of problem and deciding on the approach dependent on these patterns: 'Analyse and compare the different domains of radar data that can be exploited for objects classification, and the convenience of use of one rather than the other in different situations' and 'Propose and evaluate possible approaches for given classification problems based on radar sensors data'. Furthermore, more than half of the courses involved a larger project with a (partly) computational approach. Interestingly, whereas the ABE faculty always started out with a problem (e.g., I want to design a building and I need to know the terrain) and then took a computational approach to solve the problem, the EEMS faculty also frequently took the algorithm as a starting point and then went looking for a problem to be solved, sometimes looking at other faculties of the TU Delft to provide problems to be computationally solved.

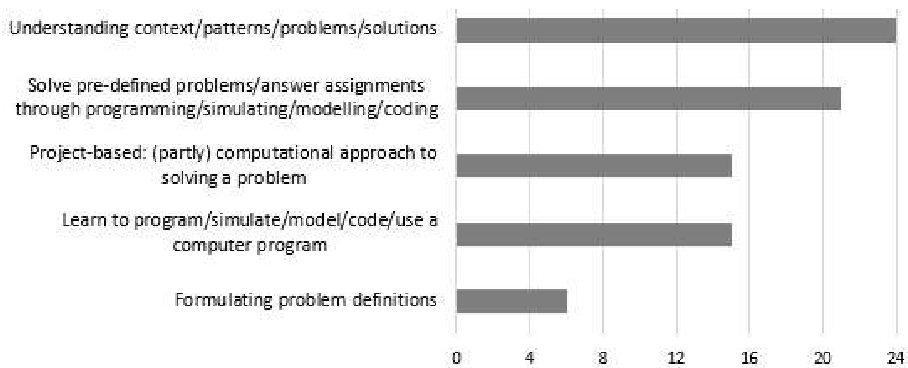

Fig. 4. EEMCS - Types of reflection of CT in courses

To conclude, these courses illustrate how CT is important also at a design faculty and the EEMCS faculty provided many CT courses. In a design faculty, there are many problems to be solved (e.g., how to best design the building) and in some of the courses a (partly) computational approach is taken to solve these problems. Meanwhile, understanding the context and patterns in problems and solutions was important in almost all courses; demonstrating the importance of this step of CT.

3.3 Educational Methods

Zooming in on the different educational methods, we see that a wide variety of methods was used in CT courses at ABE (See Figure 5). Contrary to what one might expect given the practical nature of CT, lectures were the most common instructional method. However, within one course on approximately 4 educational methods were used, signalling that lectures were often used in conjunction with different educational methods. For example, in the course Geomatics as support for energy applications three education methods are used: lectures to provide the theoretical background, practicals to allow the student to practice building models while supervision is present, and self-study to further dive into the theoretical background and do more modelling. Besides lectures, practicals and self-study, computational assignments were a common educational method, which is in line with expectations: computational assignments allow the student to apply their CT skills in practice. In addition, notable is that a fair number of courses included tutoring sessions. These sessions are a common way of teaching in the ABE faculty, during which a tutor provides feedback on the work or design of the student or students.

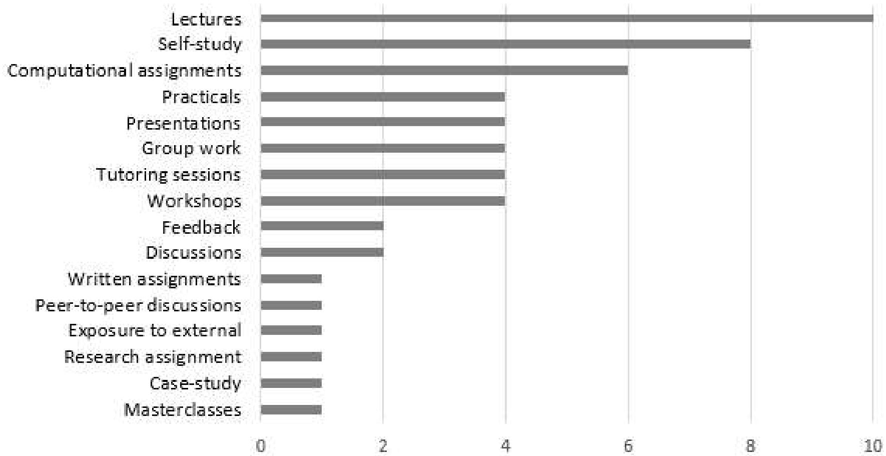

Fig. 5. ABE - Occurrences of different educational methods in CT courses

Figure 6 demonstrates the utilised educational methods at EEMCS, with on average 3 educational methods per course with lectures being the most used educational method. In 37% of the courses, practicals and/or computational assignments were used as educational methods. Larger computational groups or individual projects were more common in the EEMCS faculty. For example, in the course Crowd Computing students are working throughout the entire course for six hours per week in a group on an extensive computational project. Besides this group project, the other utilised education methods are lectures and computational assignments. Notable is the occurrence of some more innovative instructional methods like recorded lectures for the course Applied Machine learning and lectures by students for the course Machine Learning in Bioinformatics.

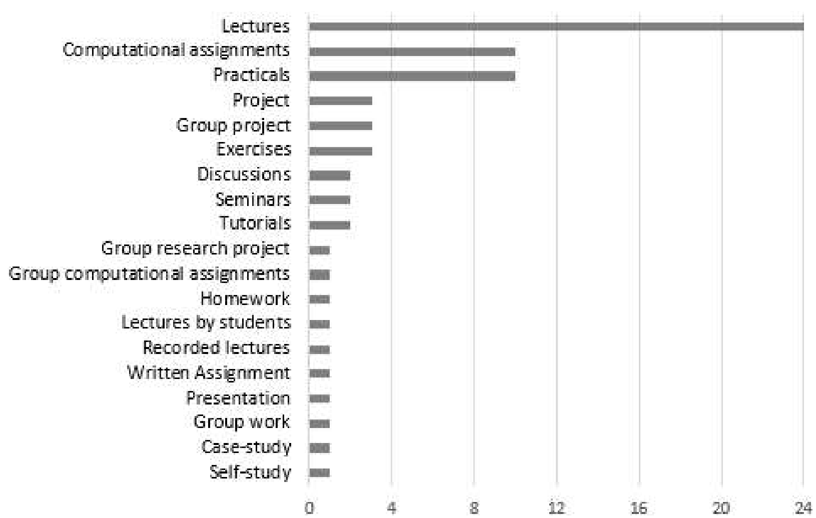

Fig. 6. EEMCS - Occurrences of different educational methods in CT courses

3.4 Assessment Methods

For the assessment at ABE and EEMCS, most courses used more than one assessment method and overview of assessment methods being used for both faculties are presented in Figure 7 and Figure 8 respectively, with computational assignment (s) being the most common assessment method. The more traditional written exam was also frequently used, often in combination with computational assignments as in e.g. Geoweb Technology, Geographical Information Systems (GIS) and Cartography and Python Programming for Geomatics. Group presentations, group reports and individual reports were also common; it should be noted that often these reports were about the developed computational model. For example, in Operations Research Methods the assessment was based on a written assignment and on a report on two mathematical models. In addition, they were often used in parallel (e.g., Algorithms for Intelligent Decision Making, Artificial Intelligence Techniques and Evolutionary Algorithms). Oral exam is also used as an assessment type; though it might be labour intensive.

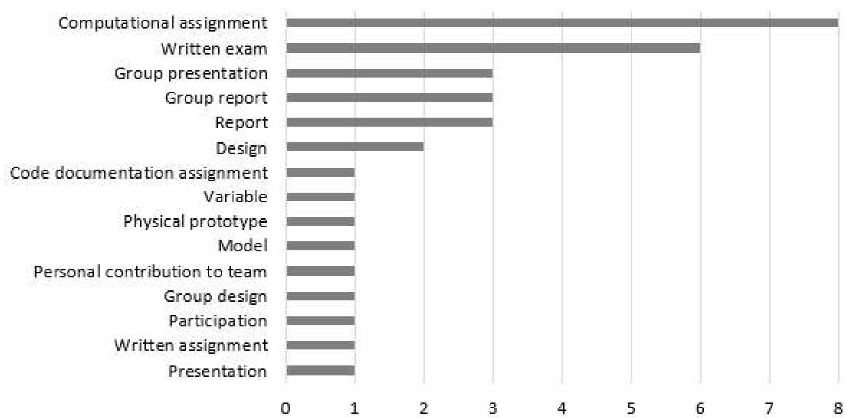

Fig. 7. ABE - Occurrences of different assessment methods.

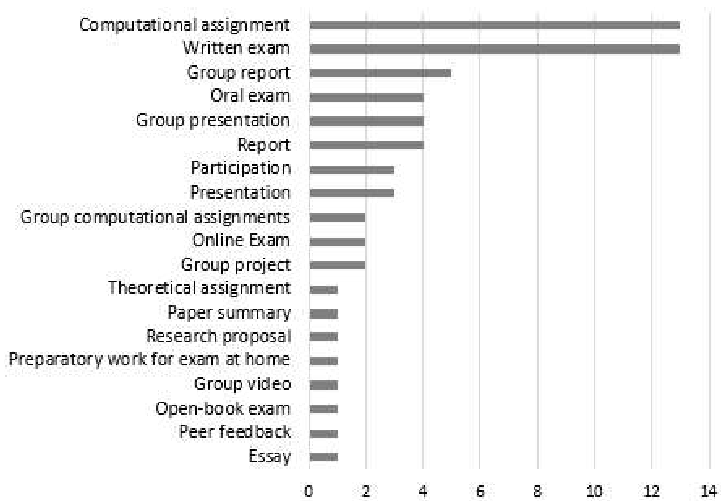

Fig. 8. EEMCS - Occurrences of different assessment methods.

4 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

With the preliminary results of this work so far, several conclusions can be made: First of all, aligned with the findings of the group concept mapping study from Specht et al. (2019), this work finds that agreement on definitions and the relations between different competences and skills under computational thinking is still vague even for experts. For developing an embedding of CT skills in an engineering curriculum necessary definitions and focus should be aligned. Secondly, the examples from the two different faculties of TU Delft show different approaches and embedding CT concepts into the curriculum. On the one hand fundamental developments or algorithms and computational abstractions need to be developed extending CS curricula and there focus, on the other hand more design oriented engineering disciplines embed computational tools and skills often in concrete design and project work. Last but not the least, we observed course designs in which computational tools and digital skills can be linked to specific subtasks of design challenges but also more generic courses which integrate computational tools as a base element of engineering design.

However, the authors spotted major limitations of this work regarding its methodology and its scope, mainly being: this is a case study on specific faculties of one university and, even though both are large and well established faculties, they might not be representative of engineering faculties in other countries. Also, that the study was based on the information on the study guide which, even if it is a requirement that it is updated yearly, might not fully reflect instruction methods that the instructors apply. Furthermore, the course descriptions used for this work was limited to academic year 2020-2021, which indicates that effectiveness of the findings in this study is time delimited and engineering education is changing with the technological and societal dynamics, which indicate that longitudinal observations and investigations are needed. Regarding the methodology, though this work follows certain level of systemacy and considers the potential bias caused by individual work, several aspects should be noted: Information coding and data synthesis was weak regarding the verification of the validity; the conclusions were made merely with observation and analysis from course descriptions, making it weak as actual situations may differ from the course descriptions.

To further advance this work, the authors plan to conduct focus group study or interviews to gain more insights about the integration of CT into Engineering curriculum and establish more solidate coding schemes and verification standards to improve the validity of the extracted information in future work.

REFERENCES

[1] Seely, B. E. (2020). Patterns in the History of Engineering Education Reform: A Brief Essay. National Academy of Engineering, Engineer of 2020: National Education (pp.114-130) Summit. National Academies Press.

[2] Kamp, A. (2016). Engineering Education in the Rapidly Changing World: Rethinking the Vision for Higher engineering Education. (2nd revised edition ex.) TU Delft, Faculty of Aerospace Engineering.

[3] Wing, J.M (2006). Computational thinking. Commun ACM. Vol.49(3):33-35

[4] Smith, M. (2016). Computer Science For All | whitehouse.gov. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all. Accessed April 20, 2021.

[5] Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

[6] Wing, J. M. (2011). Research notebook: Computational thinking—what and why. The Link Magazine, Pittsburg, PA: Carn mputer Science. Retrieved from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Accessed 10 April 2022.

[7] BBC (2018). What is computational thinking? - Introduction to computational thinking - KS3 Computer Science Revision. (n.d.). BBC Bitesize. Retrieved April 26, 2022, from https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

[8] Brennan, K. (2012). New frameworks for studying and assessing the development of computational thinking. AERA. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

[9] European Commission (2020). Joint Report of the Council and the Commission on the implementation of the strategic framework for European cooperation in education and training (ET 2020). https://eur-lex.europa.eu/legal-content/GA/TXT/?uri=CELEX:52015XG1215(02). Accessed April 20, 2021.

[10] Bocconi, S. Chioccariello, A. Dettori, G. et al (2016). Developing Computational Thinking in Compulsory Education. doi:10.2791/792158

[11] Anderson (2016). A Call for Computational Thinking in Undergraduate Psychology. Psychol Learn Teach. Vol.15(3):226-234. doi:10.1177/1475725716659252

[12] Shute, V.J., Sun, C., Asbell-Clarke, J. (2017). Demystifying computational thinking. Educ Res Rev. Vol.22:142-158. doi:10.1016/j.edurev.2017.09.003

[13] Specht, M., Coenders, M., & Stoyanov, S. (2019). Implementing Computational Thinking in the Dutch Curriculum an Exploratory Group Concept Mapping Study. CoolThink@ JC, 210.

APPENDIX

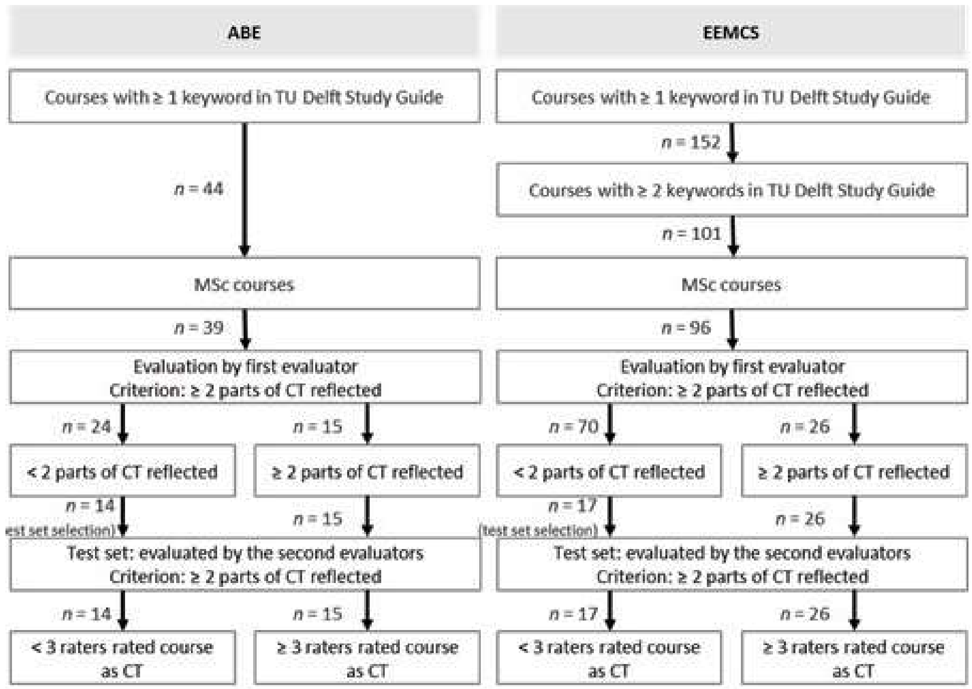

Fig. 1. Selection process of the CT courses. ABE: (Architecture and the Built Environment), EEMCS: (Electrical Engineering, Mathematics & Computer Science)

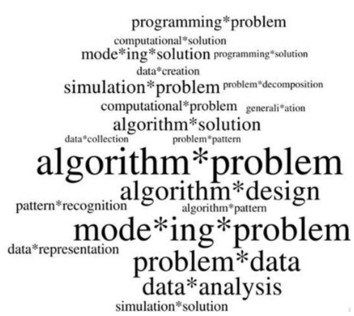

Fig. 2a. ABE – Word cloud of the keywords in the CT courses. A larger font size correlates with a higher occurrence

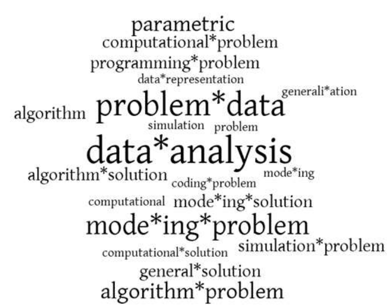

Fig. 2b. EEMCS – Word cloud of the keywords in the CT courses. A larger font size correlates with a higher occurrence

Table 1. Overview of the keywords used to identify courses that implicitly include CT

Four Components	Keywords
Problem decomposition	Problem*decomposition; Problem*data; Problem*pattern; Problem* abstraction
Pattern Recognition	Pattern* recognition; Data*analysis; Data*creation; Data*collection; Data*representation; Identif* patterns

Abstraction	Abstraction; Generali*ation; General*solution; Formulating* solution
Algorithm Design	Algorithm* design; Algorithmic* thinking; Algorithm* problem; Algorithm* pattern; Algorithm* solution; Computational* problem; Computational* pattern

Table 2. Extracted information per CT course

Categories	**Sub-categories**
General information	Date of consultation; Keyword hits; number of keywords; Course Title; Course code
Place in Curriculum	ECTS; Semester (Q1[4] Q2 Q3 Q4); Programme; Track; MSc year; Elective vs obligatory vs obligatory with choice
Study Description information	Link; Offered by; E-mail; Course content – TOPIC; Study goals – ALL; Study goals – TOPIC; Education methods; Ass. Type; Ass. Process; Ass.[5] Criteria
Evaluation	Presence of: Problem decomposition, Pattern recognition, Abstraction, Algorithm design; Type of course; Sure (yes / no); Notes

[4] Q1 – Q4 refer to Quarter 1 to Quarter 4 in an academic year.

[5] Ass.: Abbreviation of assessment.