Education 4.0 and Computer Science: a systematic review

Journal:	*Transactions on Learning Technologies*
Manuscript ID	TLT-2021-06-0146
Manuscript Type:	Regular paper
Keywords:	K.3.2.b Computer science education < K.3.2 Computer and Information Science Education < K.3 Computers and Education < K Computin, learning design, Education 4.0, Teaching competencies

Education 4.0 and Computer Science: a systematic review

Bart Rienties, Rebecca Ferguson, Dalibor Gonda, Goran Hajdin, Christothea Herodotou, Francisco Iniesto, Ariadna Llorens, Henry Muccini, Julia Sargent, Sirje Virkus, Maria Vittoria Isidori

***Abstract*— Computer science (CS) is a broad subject which is relevant in contemporary education, including Education 4.0. In this systematic literature review (SLR) we explore how CS educators implemented innovative learning designs in their teaching practices. We focus on cataloguing learning design approaches to teaching and learning of CS that are aligned with Education 4.0. The SLR included CS papers published between 2016 and 2020 from four research databases. We focused on three research questions related to I) identification of innovative pedagogic approaches used to support the teaching of CS, II) which of them are aligned with Education 4.0 and III) what skills and competencies do CS educators require in order to align their teaching with Education 4.0. 231 studies were identified of which 66 were included in the final phase. The findings indicate that none of the identified studies explicitly mentioned Education 4.0. Nonetheless, CS educators on average included 4.41 out of nine EDU 4.0 characteristics in their designs, with substantial variation (SD = 2.30). Follow-up factor analysis and k-means cluster analysis indicated that CS educators tended to design fairly consistent designs. We found a three-cluster solution: 1) EDU 4.0 light, 2) Project-based/hands-on learning, and 3) Full EDU 4.0. EDU 4.0 light. Studies who used more innovative and Full EDU 4.0 designs were more inclined to refer to skills and competences needed to teach CS. These findings suggest three broad flavors when designing innovative CS practice. Future research should explore which solutions provide more effective Education 4.0 learning experiences.**

***Index Terms*— Computer science, Learning design, Education 4.0, Teaching competencies**

I. Introduction

The way teachers design blended and online courses has a fundamental impact on how learners engage with learning activities. Under the umbrella term of learning design, a range of studies in this journal has argued that how teachers design their learning activities drives learning [1-3]. For example, in a systematic literature review (SLR) of 43 learning design studies [1] substantial growth in research and application on learning design in higher education has been noted.

Indeed in a recent special issue on learning design and learning analytics substantial progress has been made on how learning design decisions by educators impact learners in the last ten years [4]. For example, [5] examined the learning engagement of 111,256 students in 151 courses at The Open University in the United Kingdom and found that learning design choices made by educators strongly predicted engagement, satisfaction and performance of students. Nonetheless, there remains a strong need to "explore how educators plan, implement, and evaluate learning designs" [4].

In particular, there is a paucity of research within the broad discipline of Computer Science (CS) on how educators design for innovative practice [6-8]. Due to the strategic importance of CS and supporting the current and next generation of students to develop appropriate computing and data skills [6, 7, 9, 10], we specifically want to explore how CS educators are producing and implementing innovative learning designs.

This study aims to catalogue learning design approaches to teaching and learning within CS that are aligned with Education 4.0. Definitions of Education 4.0 vary but usually focus on innovation, novelty, use of technology, and connections with employment and industry [10-12]. In this article we define Education 4.0 as "an approach to learning and teaching that emphasizes the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context, and therefore requires flexible and creative approaches to its implementation" [13].

Given the contemporary conceptualization of Education 4.0 and the rapid pace of development within CS, we will review CS literature published in the last five years. Furthermore, as there is continuous change in technology and CS, we are specifically interested in whether (or not) innovative CS approaches refer to the skills CS educators need to be able to teach in an Education 4.0 manner.

II. Computer Science and Education 4.0

CS is a broad subject area that covers many disciplines and overlaps with many others. We use the definition provided by the UK Quality Assurance Agency Subject Benchmark Statement on Computing: "Computer science provides the necessary knowledge to understand and build computational systems" [14]. The statement goes on to list the main characteristics of CS and notes that, "[g]enerally, these are expressed in the ability to specify, design and write computer programs." The breadth of the field means it is able to draw on teaching methods from diverse disciplines and is also used as a way of preparing students for a wide range of professions.

This manuscript was submitted on 15 June 2021. This work was supported in part by funding from EU Erasmus+ project TEACH4EDU. Bart Rienties is the corresponding author and is with the Institute of Educational Technology, Open University UK, Milton Keynes, MK76AA, UK (e-mail: bart.rienties@open.ac.uk).

A. Previous systematic literature reviews on innovative approaches in CS

A range of SLRs on CS and innovative approaches to teaching and learning have been published in the last five years [e.g., 6, 7, 8, 15]. For example, [6] conducted a meta-analysis of 155 papers from 1980-2014, with the aim of giving "insight into the current research on the education and career development of graduates in the field of ICT". A broad range of search terms was used, and the findings in terms of curriculum design and delivery indicate a need to adjust curricula to the needs of industry. As argued by [6] "the literature indicates a need for innovative approaches in curriculum design and delivery, such as designing competency-based programs that are not restrained by the traditional semester seat-time model, providing flexible curriculum and minimising the time spent in the classroom, or offering courses on not only how to manage innovation, but also on how to innovate". In terms of teaching, methods that could help to encourage graduate employability include learning by doing, learning from mistakes, team-work and collaborative learning. Furthermore, a job-oriented experiment course system; problem- or project-based learning; and work-integrated learning to develop a wide range of desired skills, competences and knowledge are encouraged, which link well with some of the Education 4.0 concepts.

In a review of 157 learning designs implemented at The Open University, [15] found that the majority of educators primarily used two types of learning activities, namely assimilative activities (e.g., reading, watching videos, listening to audio) and assessment activities. Often educators combined assimilative, productive (e.g., coding) and assessment activities or, alternatively, assimilative, finding and handling information and communication tasks (e.g., working together with peers).

Using a SLR of 34 papers how software engineering education was aligned with industrial needs in the period 1995-2018, [7] identified eight research questions, of which two are highly relevant to our project (What curriculum models have been used to design the studies?; What educational recommendations are provided in each study?). [7] indicated that the "qualitative coding provided four themes for the educational recommendations provided in the papers: (1) Need for more emphasis on soft skills (20 papers), (2) Need for active Infrastructure as Code (IAC) (3 papers), and (3) Less emphasis on certain topics (2 papers))." In order to encourage development of soft skills, educators need to use real-life projects, implement industry-academia collaboration in the design of education, and anticipate future trends, while also preparing students to deal with those trends [7].

Based upon 195 empirical papers [8] provided an overview of the practices that have been used to integrate software testing into programming education. The study showed that testing practices in programming assignments involved students to different extents: analysing test results from submission tools, working with instructor-provided tests, using support mechanisms to design tests (e.g. plugins where students insert inputs and expected outputs) and, finally, students writing their own tests. However, few studies addressed how students learned testing concepts in programming courses [8].

While these SLRs provide important and deep insights into how CS, computer programming, and softer skills have been used in a range of CS contexts, none of these studies specifically focus on, mention, or include concepts of Education 4.0. Furthermore, none of these reviews specifically looked at the way the respective learning designs were used.

B. Education 4.0

This SLR catalogues approaches to teaching and learning within CS that are aligned with Education 4.0. This is a relatively new term – Harkins originally proposed it in 2008 to describe innovation-producing education [16] as opposed to knowledge-producing education. Definitions of Education 4.0 vary but usually focus on innovation, novelty, use of technology, and connections with employment and industry [10-12]. The number 4.0 makes a connection with the view that there have been four industrial revolutions with the current Industry 4.0 increasingly automated, making use of modern smart technologies and the Internet of Things (IoT).

As the meaning of what Education 4.0 means is still being negotiated, this SLR uses two conceptualizations of Education 4.0. Fisk [17] and later on Hussin [11] identified nine characteristics associated with EDU 4.0:

1 Learning any time / anywhere: Students will be able to learn where and when they choose.

2 Personalized learning: Study tools will adapt to the capabilities of the student.

3 Choice how to learn: Students will be able to modify their learning process.

4 Project-based learning: Students will learn to apply their skills in a variety of situations.

5 Hands-on learning: Students will have authentic experiences and gain real-world skills.

6 Data interpretation: Students will learn to interpret and reason with data.

7 Assessed differently: Knowledge and skills will be assessed in new ways.

8 Student ownership of curriculum: Students will have critical input into their courses.

9 More independent: students will become more independent.

The elements identified are all potentially innovative. However, they focus on students rather than on the broader picture of how innovations are developed and embedded in terms of learning design. As indicated in the introduction, in this study we propose an alternative definition of Education 4.0 that draws on ideas and descriptions in a range of literature [10, 12, 18-22]. We define Education 4.0 as an approach to learning and teaching that emphasizes the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context, and therefore requires flexible and creative approaches to its implementation. Note that in the remainder of this study, when we use EDU4 we refer to the Hussin [11] conceptualization, while Education 4.0 refers to our own definition.

As evidenced by a range of studies, being able to design and implement innovative pedagogical approaches requires substantial new and/or updated skills and competences from educators to make use of Education 4.0 approaches. A recurring theme seems to be a shift from CS teachers as being a knowledge transmitter to a teacher as a facilitator or moderator or consultant of learning [23]. Teachers could achieve that by being flexible (adapt to change) [24], supportive, help students to develop ownership of learning [25], foster an environment where students take risks and share what they do not know about, and where failure is acceptable. This role was often discussed within a flipped classroom implementation [25] that could give control to students to study the teaching material at their own pace and contact the teacher to solve problems and discuss their learning. In such conditions, the teacher is monitoring a student's progress and facilitates understanding through discussions [26]. An increasing number of CS teachers have started to implement project-based learning and hands-on experiences in their classroom [27]. However, the specific skills and competences needed to design, implement and evaluate effective Education 4.0 CS courses has received limited attention.

In particular with the COVID-19 pandemic and the rapid shift to online education, it is essential to update our insights as to how CS teachers are adopting innovative pedagogies and Education 4.0 approaches. Obviously, what is innovative or not is dependent on the context. We acknowledge that a pedagogy or technology that is really innovative in institution 1 might be common practice in institution 2, and vice versa. In order to investigate which innovations are being introduced in the field of CS a SLR was carried out, focusing on three research questions:

RQ1: Which innovative pedagogic approaches are used to support the teaching of Computer Science (CS)?
RQ2: Which of these approaches align with Education 4.0?
RQ3: What skills and competences do CS educators require in order to align their CS teaching with Education 4.0?

III. Method

In this SRL we follow recommendations from [1, 28]. Four research databases were searched: Science Direct, Wiley InterScience, Web of Science, and Scopus. These were chosen because of their ranking as academic research databases, and good coverage of studies relevant for the review. Papers had to be published in English during the five-year period 2016–2020, thereby increasing the chance that a particular study used a contemporary and innovative pedagogical approach in CS. Keywords had to include Computer Science; undergraduate and/or postgraduate; as well as education, teaching and/or pedagogy. The following search string was used: “computer science” AND education AND teaching AND pedagogy AND ("undergraduate" OR "postgraduate"). These search terms identified 231 unique publications across the four databases. Publications identified using the search criteria were excluded if any of the following exclusion criteria applied: 1) The focus was on primary and/or secondary education; 2) The focus was on a subject other than Computer Science; 3) The focus was on learners (e.g., their gender or expectations) rather than teaching.

A. *Coding process*

Applying the inclusion and exclusion criteria, 231 studies were identified. In Phase 0, Author RF manually screened the abstracts to check whether the respective studies should be included or excluded based upon the above criteria. Subsequently, 75 studies were excluded.

In Phase 1, following a one-hour online training and discussion of the online coding scheme, 156 studies were read in depth by 18 members of TEACH4EDU project and categorized based upon three inclusion criteria (i.e., 1) is it an ‘innovative’ application in a CS course; 2) Does it use technology or pedagogy in an innovative way; 3) Is the innovation evaluated, if so how?).

By including experts in CS and educational technology from six EU countries we aimed to develop an inclusive multi-disciplinary team of coders to analyze the literature and the innovative nature of the pedagogy used in a CS context. On average the members coded 8.26 studies (range: 3-11), whereby 68 studies were included for subsequent analysis. All studies were annotated and uploaded in Google Drive for a second round of coding.

In Phase 2, 17 members of the TEACH4EDU project participated in a follow-up one-hour online training and discussion of the online coding scheme of 20 variables. Coders were randomly allocated a new set of studies to code in comparison to their initial coding in Phase 1, thereby ensuring that at least two coders checked and independently coded each “innovative” pedagogy in CS. For RQ2, we adopted the nine key EDU 4.0 characteristics [11] and our Education 4.0 definition. For the analysis we used both the individual EDU 4.0 scores as well as the aggregate score. For RQ3, coders indicated whether (or not) any specific skills required by teachers to support the teaching of CS to students were mentioned. If yes, coders could use a follow-up open text box to add any description and conceptualization of teacher skills. Following this, we recoded and aggregated the skills.

On average 4.25 studies (range: 2-10) were coded per coder based upon the coding scheme developed from the above research questions. A random sample of 15 studies was double coded and indicated reliable coding (average Cohen Kappa EDU 4.0 = 0.84). Afterwards, the first coders from Phase 1 checked the codes from the second coders in Phase 2, discussed any differences, and agreed on the final coding (average Cohen Kappa EDU 4.0 = .93). If a study did not indicate any EDU 4.0 characteristic, we removed it from further analysis, and therefore we ended up with a total of 66 studies.

B. *Data analyses*

The vast majority of studies included referred to undergraduate CS students (79%), followed by a mix of undergraduate and post-graduate students. Five studies did not explicitly mention the specific student population, and one included teachers only. 36% of studies were from the USA, followed by Spain (9%), Brazil (8%), and Germany (6%). Using the GLOBE geo-cultural regions classification [29], 47% of studies were conducted by Anglo-Saxon countries, followed by Latin American countries and Latin European (each 12%), Eastern European (8%), Scandinavian and German countries (each 6%), Confucian Asian (5%) and Middle Eastern countries (3%). No studies were identified from African or Southern Asian countries. Using ANOVA analyses, no significant differences were found on our key variables and GLOBE, indicating no substantial differences in practices in CS based upon national/geo-cultural regions.

In terms of reporting the findings of RQ1-2, we first explored the overall data, then carried out an exploratory factor analysis (principal component analysis) with direct oblimin rotation to identify a common structure in the EDU characteristics. Multiple factor structures were explored, but a two-factor structure had the best fit. Finally, a k-means cluster analysis was conducted in order to explore any common patterns in terms of learning designs employed by CS educators. For RQ3 all articles were screened whether (or not) reference was made towards teachers' competences and skills to implement a respective innovation. If a study explicitly mentioned this, this was coded and included in an open text box. These open text boxes were afterwards analyzed by authors CH and JS to find common patterns.

IV. Results

In terms of RQ1 and RQ2, of the 66 studies selected perhaps surprisingly none of the studies explicitly mentioned "Education 4.0". In part, this could be a result of the relatively recent conceptualization of Education 4.0, and in part this could be due to the lack of adoption of the term Education 4.0 in the discipline of CS. Based upon the coding scheme, 66 studies included at least one EDU 4.0 characteristic [11]. Furthermore, in total 54 articles (80%) were considered to fit our own Education 4.0 definition.

As indicated in Figure 1, on average the 66 studies included 4.41 of the nine EDU 4.0 characteristics, with a substantial variation (SD = 2.30). There seemed to be two peaks in Figure 1, whereby 35% of studies only had 2-3 EDU 4.0 characteristics, while another peak at 7 EDU 4.0 characteristics was present.

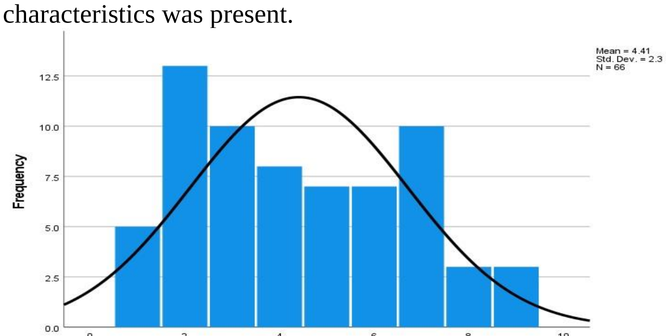

Figure 1 Histogram of Education 4.0

The most common EDU 4.0 characteristic was 5) hands-on learning (73%), followed by 9) more independent (67%), 4) project-based learning (61%). Around half of the studies included the characteristic that 1) learning any time / anywhere, while around a third of studies included 7) assessed differently (35%) and 8) student ownership of curriculum (32%). Furthermore, as illustrated by the error bars, there was substantial variation in the 66 CS practices.

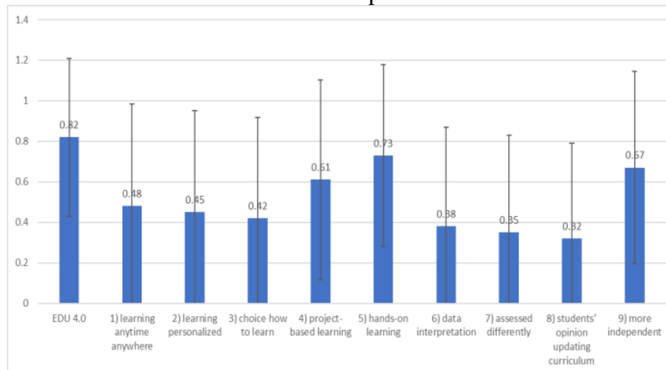

Figure 2 EDU 4.0 characteristics (with error bars)

We found a moderately strong correlation (rho = .429, p < .01) between the aggregate Hussin [11] and our Education 4.0 definition, with the strongest correlation on the EDU 4.0 characteristic 5 (rho = .417, p < .01). The individual EDU 4.0 characteristics were not all directly and significantly correlated. Therefore, a factor analysis was conducted on the data collected, which indicated the existence of two factors with item loads of .45 and more. The first component had an eigenvalue of 2.62 (corresponding to 29% of the explained variance), the second component had an eigenvalue of 1.45 (corresponding to 16% of the explained variance). As indicated in

Table 1, EDU 4.0 characteristic 2, 1, 6, 9, and 3 loaded on the first factor, which we will label as "individual choice and development". EDU 4.0 characteristic 4, 5 and 7 loaded on the second factor, which we will label as "intention project-based/hands-on learning". EDU 4.0 characteristic 8 did not load on any factor. Respective Cronbach Alphas for these two factors were .68 and .62, indicating reasonable reliability. In other words, teachers often combined EDU 4.0 characteristics together when designing and implementing CS courses based upon these two factors.

Table 1
Pattern structure of Factor analysis EDU 4.0 Characteristics

	1	2
2) learning will be personalized to individual students	.769	
1) learning can take place anytime anywhere	.644	
6) students will be exposed to data interpretation in which they are required to apply their theoretical knowledge to numbers and use their reasoning skills to make inferences based on logic and trends from given sets of data	.618	
9) students will become more independent in their own learning	.617	
3) students have a choice in determining how they want to learn	.589	
8) students' opinion will be considered in designing and updating the curriculum		
4) students will be exposed to more project-based learning		.883
5) students will be exposed to more hands-on learning through field experience (e.g., internships, mentoring projects, collaborative projects)		.836
7) students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient		.455

Extraction Method: Principal Component Analysis.

Rotation Method: Oblimin with Kaiser Normalization.

Finally, a follow-up analysis using k-means cluster techniques indicated a three-cluster solution across the 66 studies. As illustrated in Figure 3, there seemed to be three clusters of studies, which we label as 1) *EDU 4.0 light* (n = 18), 2) *project-based/hands-on learning* (n = 22), and 3) *full EDU 4.0* (n = 26). With the notable exception of EDU 4.0 characteristic 8, using ANOVAs all EDU 4.0 characteristics were significantly different between the three clusters with large effect sizes. In other words, there appeared to be three distinct innovative pedagogical practices present in published work on CS in the last five years.

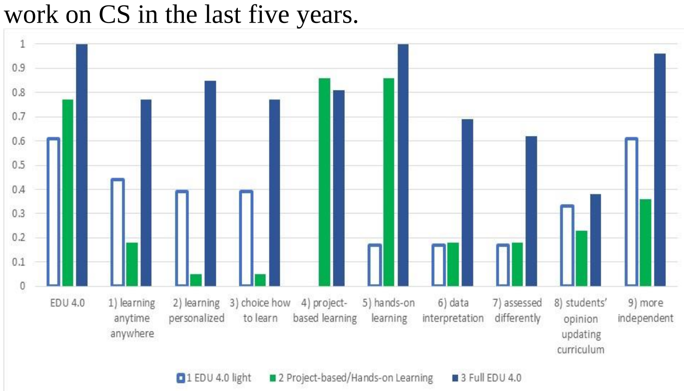

Figure 3 Cluster analysis of EDU 4.0 (3 cluster solution)

As indicated in Figure *4*, EDU 4.0 light studies (blue circles) mostly had relatively low total EDU 4.0 scores, and often did not include project-based activities. Therefore, most of these studies in Figure *4* were positioned on the bottom left. In contrast, while some project-based/hands-on learning studies (green circles) also had relatively low EDU 4.0 scores, in particular for personalized learning and choice how to learn, they had a strong focus on project-based and hands-on learning. Therefore, many of these studies are positioned in the middle to top-left quadrant of Figure *4*. Finally, studies which were classified as full EDU 4.0 studies (blue triangles) were mostly positioned in the middle and right of Figure *4*, indicating these studies used more and even all EDU 4.0 characteristics in their designs. Note that the numbers in Figure 4 refer to the studies discussed below.

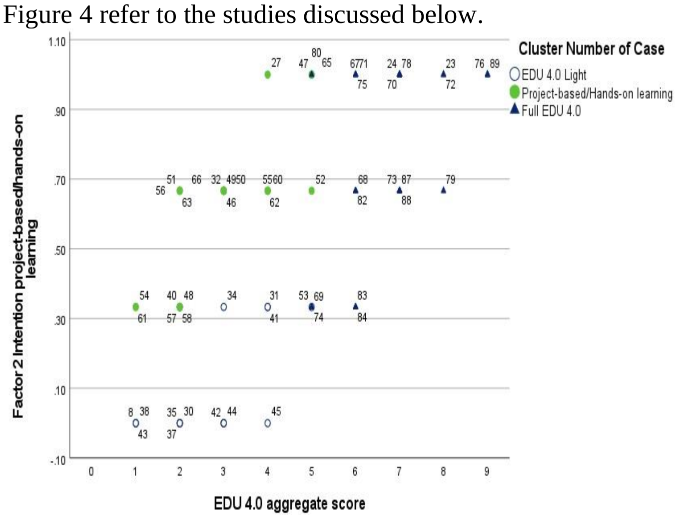

Figure 4 Scatterplot of cluster analysis results of 66 studies

A. *EDU 4.0 light*

As indicated in Table 2, in EDU 4.0 light studies teachers mostly focused on more 9) more independent (61%), learning any time / anywhere (44%), personalized learning (39%), and choice how to learn (39%), but with limited hands-on learning (17%) and no project-based learning (0%). For example, [30] introduced the concept of a modern C ++ course for students of electrical engineering and CS based on an inverted classroom and with attractive IoT hardware. The main goal of the new course was to reduce lecture time in favor of practical learning of students through programming.

TABLE 2
EDU 4.0 LIGHT STUDIES

Authors	E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	Country
Apiola, Lokkila and Laakso [31]	Y		Y		Y				Y	Finland
Burrows and Borowczak [32]	Y	Y		Y				Y	Y	USA
Degener, Haak, Gold-Veerkamp and Abke [33]	Y								Y	Germany
Dickson, Dragon and Lee [34]		Y	Y		Y					USA
Dondio and Shaheen [35]								Y	Y	Ireland
Fisher, Rader and Camp [36]	Y	Y						Y		USA
Frevert, Rorrer, Davis, Latulipe, Maher, Cukic, Mays and Rogelberg [37]								Y	Y	USA
Giacaman and De Ruvo [38]			Y							New Zealand
Hosseini, Hartt and Mostafapour [39]							Y		Y	USA, Wales, Canada
Parejo, Troya, Segura, del-Río-Ortega, Gámez-Díaz and Márquez-Chamorro [40]	Y						Y			Spain
Park and Kim [41]	Y	Y					Y		Y	Korea
Pilkington [42]	Y					Y			Y	South Africa
Scatalon, Garcia and Barbosa [8]		Y								Brazil
Schäfer [30]	Y								Y	Germany
Shi, Min and Zhang [43]			Y							China
Silva, Steinmacher and Conte [44]			Y					Y	Y	Brazil
Tyler and Abdrakhmanova [45]		Y	Y			Y		Y		Kazakhstan

Note that E1-E9 refer to EDU 4.0 characteristics.

Another course [33] integrated LEGO MindStorms EV3 robots within lessons teaching the programming language ANSI-C. The intention was to make practical programming lessons more tangible and closer to the future field of work for CS and engineering students. Although the EV3 contributed to learning success and fun during the lessons, students were not able to program the EV3 outside laboratory opening hours and practical lessons. To solve this problem, a simulation was provided to make the programming task time- and location-independent. Another case [40] presented a flipped course on software architecture and integration that formed part of a Software Engineering degree. In addition, the gamified platform Kahoot was used for interactive tests at the beginning of the laboratory sessions. Based on responses to these quick quizzes, the lab instructor decided which concepts to clarify. Students had, on average, 24 more minutes per session to solve in-class exercises when using the flipped-classroom approach. More than 70% of students considered the quantity, duration and didactic content of the videos (very) appropriate; and 90% of students preferred this approach for laboratory sessions. While each of these studies indicated substantial innovative pedagogical enhancements, most of these studies focussed only on some of the EDU 4.0 characteristics.

B. Project-based/hands-on learning

The second cluster , which we labelled as project-based/hands-on learning, had a strong focus on project-based learning (86%) and hands-on learning (86%), with relatively limited focus on choice how to learn (5%), personalized learning (5%), and learning any time / anywhere (18%), as illustrated in Table 3. For example, [46] assessed the benefits of the use of technology and active learning practices (i.e., Project-Based Learning and Peer Instruction) in the classroom with 25 students to contribute to a more effective and efficient learning environment. Another example [27] described the use of an innovative platform to improve the knowledge of 51 CS students about software testing by providing a set of learning objects and tutorials categorized by difficulty level. This evolved into a collaborative learning environment that included social networking features such as the ability to award virtual points for student social interaction about testing [27].

TABLE 3
PROJECT-BASED/HANDS-ON LEARNING STUDIES

Authors	E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	Country
Aghaee and Keller [47]	Y			Y	Y		Y		Y	Sweden
Alasbali and Benatallah [48]				Y					Y	Global
Alegre, Moreno, Dawson, Tanjong and Kirshner [49]			Y	Y	Y					USA
Alomari, Ramasamy, Kiper and Potvin [27]	Y			Y	Y		Y			USA
Berikan and Özdemir [50]				Y	Y	Y				Turkey
Bielefeldt, Polmear, Swan, Knight and Canney [51]				Y	Y					USA
Borowczak and Burrows [52]				Y	Y	Y		Y	Y	USA
Burrows and Borowczak [53]					Y		Y	Y		USA
Bushmeleva and Baklashova [54]					Y					Russia
Caceffo, Gama and Azevedo [46]				Y	Y				Y	Brazil
Carrascal, del Barrio and Botella [55]				Y	Y	Y		Y		Spain
Casañ, Alier and Llorens [56]				Y	Y					Spain
Chamberlin, Hussey, Klimkowski, Moody and Morrell [57]	Y				Y					USA
Cobos and Roger [58]				Y				Y		Spain
Fagerholm, Hellas, Luukkainen, Kyllönen, Yaman and Mäenpää [59]				Y	Y				Y	Finland
Juárez, Aldeco-Pérez and Velázquez [60]		Y		Y	Y				Y	Mexico
Lewis and Lacher [61]				Y						USA
Liang and Chapa-Martell [62]	Y			Y	Y				Y	Japan
Llorens, Berbegal-Mirabent and Llinas-Audet [63]				Y	Y					Spain
Mäkiö, Yablochnikov, Colombo, Mäkiö and Harrison [64]				Y	Y				Y	UK
Santos, Dischler, Adzhiev, Anderson, Ferko, Fryazinov, Ilčík, Ilčíková, Slavik, Sundstedt, Svobodova, Wimmer and Zara [65].				Y	Y	Y	Y	Y		Austria, Czech Republic, Slovak Republic, UK
Seyam and McCrickard [66]				Y	Y					USA

[56] provided a critical review of 29 years of teaching courses on social, environmental, and ethical issues to students of Informatics Engineering in Spain. Strategies included case study sessions and active methodologies. Collaborative approaches included the jigsaw method, think-pair-share, group investigation, and role-playing debates in online forums. Over time, the use of wikis to support collaboration has given way to use of Google Drive. [59] implemented a course where students were considered as prospective entrepreneurs, as well as potential employees in modern, start-up-like intrapreneurship environments within established companies. This paper reported on experiences gained during seven years of teaching start-up knowledge and skills, whereby a Software Factory, an educational environment for experiential, project-based learning, was developed. However, due to the nature of project-based learning in nearly all of these studies there was relatively low flexibility in terms of anytime/anywhere, personalisation, and choice of study.

C. Full EDU 4.0

The third and final cluster, which we labelled as the full EDU 4.0 version, was strongly focused on hands-on learning (100%), more independent (96%), personalized learning (85%), learning any time / anywhere (77%) and choice how to learn (77%). The lowest EDU 4.0 characteristic was student ownership of curriculum (38%), as illustrated in Table 4, although this was substantially higher than the other two clusters.

TABLE 4
FULL EDU 4.0 STUDIES

Authors	E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	Country
Alsaif, Li, Soh and Alraddady [67]	Y		Y	Y	Y		Y		Y	Saudi Arabia
Behnke, Kos and Bennett [68]	Y		Y	Y	Y	Y			Y	USA
Borge, Ong and Goggins [23]	Y	Y		Y	Y	Y	Y	Y	Y	USA
Broisin, Venant and Vidal [69]	Y	Y	Y		Y				Y	France
Buffardi and Valdivia [70]			Y	Y	Y	Y	Y	Y	Y	USA
Charlton and Avramides [71]		Y	Y	Y	Y		Y		Y	UK
Corritore and Love [25]	Y		Y	Y	Y		Y	Y	Y	USA
Gestwicki and McNely [24]		Y	Y	Y	Y		Y	Y	Y	USA
Gonçalves, von Wangenheim, Hauck and Zanella [72].	Y	Y	Y	Y	Y	Y	Y		Y	Brazil
Goumopoulos, Nicopolitidis, Gavalas and Kameas [73]	Y	Y	Y	Y	Y	Y			Y	Greece
Knobelsdorf, Frede, Böhne and Kreitz [74]	Y	Y			Y	Y			Y	Germany
Munkvold [75]		Y	Y	Y	Y		Y		Y	Norway
Paschoal, Oliveira, Nakagawa and Souza [76]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Brazil
Pawelczak [77]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Germany
Peng [78]	Y	Y	Y	Y	Y	Y	Y			USA

Peteranetz, Flanigan, Shell and Soh [79]	Y	Y	Y	Y	Y	Y		Y	Y	USA
Pivkina [80]		Y		Y	Y		Y		Y	Mexico
Ruiz, Serral Asensio and Snoeck [81]	Y	Y		Y	Y	Y	Y		Y	Cuba
Salem, Damaj, Hamandi and Zantout [82]	Y	Y		Y	Y	Y			Y	Lebanon
Seyam, McCrickard, Niu, Esakia and Kim [83]	Y	Y	Y		Y	Y			Y	USA, Korea
Silva, Polo and Crosby [84]		Y	Y		Y	Y		Y	Y	USA
Tanaka, Ferreira da Silva and Casanova [85]	Y	Y		Y	Y	Y	Y		Y	USA
Tlili, Essalmi, Jemni and Kinshuk [86]	Y	Y	Y	Y	Y	Y			Y	Tunisia
Troussas, Krouska and Sgouropoulou [87]	Y	Y	Y		Y		Y	Y	Y	Greece
Winiecki and Salzman [88]	Y	Y	Y	Y	Y	Y			Y	USA
Wood, Clements, Peterson, Janzen, Smith, Haungs, Workman, Bellardo and DeBruhl [89]	Y	Y	Y	Y	Y	Y	Y	Y	Y	USA

For example, [80] described the experience of using an undergraduate student as a peer learning assistant (PLA), supporting 80 students in in three different undergraduate computer science courses with 20 students. PLAs held office hours, helped with the labs and tutorials and facilitated student group work in class. They were therefore practising tutoring their peer students. The impact of the experience was measured by comparing student interactions between PLAs and regular university teaching assistants, showing a preference of students to interact with their PLAs. While that research offers a case for practical learning it does not offer any choice of learning anytime or anywhere, limiting the experience to the interaction with the PLAs.

A similar case was found in [83] who evaluated whether pair programming (an agile software development practice, used in both industry and education, which enforces a role-based approach to learning new programming concepts) would help 53 students during five sessions in a mobile development course to better understand mobile programming. For the evaluation of the experience, observations and questionnaires were used to show a rich experience where programming for mobile devices goes beyond merely writing code. This study shows a practical hands-on learning case for the students in a real-life software development environment, but students did not have the option to choose the way to learn or personalize their learning experience. Another case [46] assessed the benefits of the use of active learning practices teaching algorithms, data structures and programming logic in a CS introductory course with the feedback of two instructors and 24 undergraduate students via interviews and surveys to contribute to a more effective and efficient learning environment. The study indicated that both students and instructors enjoyed the use of new technologies and active learning in the course, but they would like to prioritize two-way communication between students and instructor, and collaboration among students during class.

Active learning strategies based on students' practical work combined with continuous feedback (such as the Inspection-based strategy based on doing and reflection) are preferred by students in their education. According to [84] using these active learning strategies was not appropriate at the beginning of a course, as their use could confuse students. Students need to have prior knowledge about the content to use active learning strategies.

[73] addressed distance-education challenges through advanced educational material, intelligent tutoring systems, and virtual laboratories. Students engaged in small-scale projects and implemented both software and hardware prototypes. In a flipped classroom approach [77] found that students seemed more motivated when they could work with the course material at times of their choosing, and that they were better prepared in the flipped classroom and discussions could be established on a higher level. However, the effort involved in setting up the flipped classroom was very high and course materials had to be updated frequently as programming languages evolved. Another example illustrated an intelligent mobile game-based learning application in a HE course to assess and advance learners' knowledge of the programming language C# [87]. The application employed a knowledge-assessment module to test the knowledge of learners, a recommendation module to propose personalized collaboration, a dynamic fuzzy logic-based advice generator for tailored assistance to learners' profile and misconceptions, and a cognitive learner modeler that supported the other modules. [87] concluded that incorporating personalisation and collaboration in mobile game-based learning can help students increase their knowledge level.

V. Skills for Teachers in Computer Science to Deliver EDU 4.0

In terms of RQ3, nearly half the studies (n = 30) reviewed made an explicit reference to skills and competences CS educators should have or develop to align their CS teaching to Education 4.0. As indicated in Figure 5, studies that referred to skills and competences of teachers on average had higher scores on nearly all EDU 4.0 characteristics, with the notable exception of 6) data interpretation. A follow-up ANOVA analysis indicated significant differences between studies that did and did not mention skills of teachers on 3) choice how to learn ($F = 4.758$, $p < .05$); 5) hands-on learning ($F = 5.689$, $p < .05$); 8) including students' opinion in updating curriculum ($F = 5.922$, $p < .05$); and 9) more independent learning ($F = 7.422$, $p < .01$).

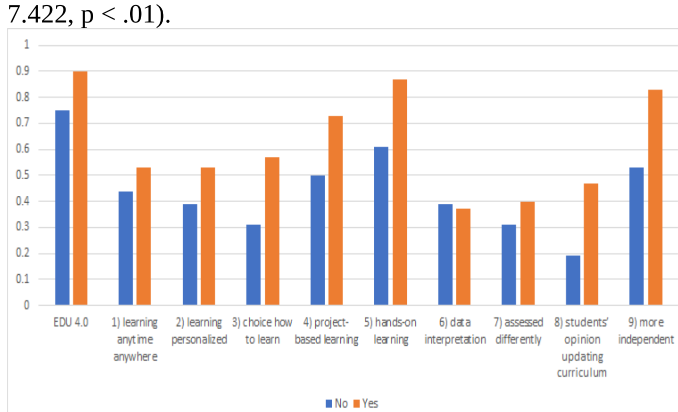

Figure 5 References to skills and competences of teachers and Education 4.0

Furthermore, the aggregate EDU4 score was substantially higher in studies that mentioned skills and competences of teachers (M = 5.30, SD = 2.20, F = 9.304, $p < .01$) relative to those who did not mention those skills (M = 3.67, SD = 2.14). Although no significant differences across the three clusters were found between studies that did or did not mention skills and competences of teachers, 53% of studies who did were part of the full EDU 4.0 cluster, while only 28% of studies who did not were part of that cluster. In other words, those studies that explicitly referred to skills and competences for teachers seemed to be more explicit and innovative in terms of pedagogies and EDU 4.0 elements. An alternative explanation could be that the authors who employed more innovative pedagogical approaches and full EDU 4.0 modes provided more narratives about how teachers could effectively support these innovative approaches.

The studies that made an explicit reference to skills and competences of HE educators discussed the issue of educators' skills in relation to the implementation and assessment of an innovative learning intervention, which was the main focus of the article. A reference to or discussion of skills and competences was often presented as an implication of the proposed study rather than being examined as the starting point of a given article. This could be explained by the fact that innovative teaching approaches or interventions are more likely to require teachers to develop new skills and competences, and thus such a discussion was seen as very relevant. This observation could explain some of the insights of the quantitative analysis, in particular the observed higher scores on the EDU 4.0 characteristics in studies where skills and competences of teachers are discussed.

In terms of the educators' skills discussed in these studies, some of them could be seen as generic, such as the creation of student-centered environments. Others were more concrete, such as the use of specific mobile games in teaching. A recurring theme we identified was the teacher as facilitator, moderator or learning consultant [23, 24, 60] as opposed to a teacher controlling or being the center of the learning process [25]. Teachers could achieve that by being flexible (adapting to change) [24], supportive, helping students to develop ownership of learning [25], fostering an environment where students take risks and share what they do not know about, and where failure is acceptable [23].

This facilitative role was often discussed within a flipped classroom implementation [25] that could give control to students to study the teaching material at their own pace and contact the teacher to solve problems and discuss their learning. In such conditions, the teacher was monitoring student progress and facilitating understanding through discussions [76]. A teacher as facilitator was also seen as the person strengthening communication, ethics, leadership, security, and software skills [60]. These conditions point to teachers as the agents in charge of developing student-centered learning environments [87].

Teachers' skills and competences were also discussed in relation to the development of more specific expertise, including the use of social network analysis techniques to understand social relationships when students are part of an online network or community [23]. Furthermore, several articles referred to the use of a peer learning assistance approach, that is, having peers to hold office hours, help with labs and facilitate student group work, as they were shown to better support learning than teaching assistants [80], and the use of specific educational games [70, 87] and remote laboratories [69] that could support CS education. In terms of game-based approaches to CS, teachers should have skills to provide tailored and personalized feedback [87] and assign students to game roles within a course management system [70].

VI. Discussion

This systematic literature review (SRL) used a three-phase coding process to review 66 studies selected from an initial data search of 231 studies in order to identify common pedagogical approaches, aligned with Education 4.0, used to support teaching computer science (CS) courses. In terms of RQ1 and RQ2 none of the 66 studies included explicitly mentioned “Education 4.0”. This could be a result of the recent conceptualisation of Education 4.0, or due to a lack of adoption of the term Education 4.0 in the specific discipline of CS.

A k-means cluster analysis indicated a three-cluster solution. EDU 4.0 light studies mostly had relatively low total EDU 4.0 scores, and often did not include project-based activities. EDU 4.0 light studies mostly focussed on more independent, learning any time / anywhere, personalized learning, and choice how to learn. As illustrated by the descriptions of these studies, substantial technological and pedagogical innovations were introduced in CS courses, although mostly focused on just one or two EDU 4.0 characteristics. This could be linked to teachers being willing to make some innovations based upon a particular problem perceived in a course, but “updating” parts of the pedagogy rather than fully redesigning a CS course [1, 6, 90].

The second cluster, which we labelled project-based/hands-on learning, had a strong focus on project-based learning and hands-on learning. These studies mainly used collaborative and project-based learning approaches with some interesting innovations, such as where CS students were considered as prospective entrepreneurs [59]. In all 21 studies there was a strong focus on hands-on and project-based learning, allowing CS graduates to develop strong programming and soft skills, often working in teams. However, due to the nature of project-based learning there was relatively low flexibility in terms of anytime/anywhere, personalisation, and choice of study.

The third and final cluster, Full EDU 4.0, was strongly focussed on hands-on learning, more independent, personalized learning, learning any time / anywhere, and choice how to learn. The lowest EDU 4.0 characteristic was student ownership of curriculum, although this was substantially higher than the other two clusters. Several innovative and integrated perspectives were used including flipped classrooms [77], game-based learning [87] and online lab work [69], indicating how CS teachers might help students to develop strong project, programming, and team skills.

In terms of RQ3, about half of the studies made an explicit reference to skills and competences CS educators should have or develop to align their CS teaching to Education 4.0. Perhaps interestingly, those studies that did refer to skills and competences of teachers on average had significantly higher scores on nearly all EDU 4.0 characteristics. This might indicate that CS authors who employed more innovative pedagogical approaches, in particular when implementing flipped classrooms or interactive games or lab-exercises, felt the need to provide more detailed narratives about their peer teachers needed to be aware of the need for additional skills and competences to implement these innovative approaches. Learning practices associated with Education 4.0 require considerable time for preparation compared to the traditional lecture-based class [46] and this may be overwhelming for teachers. Therefore, support should be provided through, for example, teaching assistants, fellow teachers, or the reuse of existing activities to help teachers gradually develop the proposed skills and competences.

Based on the research studies reviewed in this study, we can conclude that Education 4.0 is a new concept in teaching Computer Science courses and has not yet been utilised by teachers. This study indicated that although this field is at its early beginnings, some basic trends can be noted and conceptualised. In a way, it was surprising to identify three clear clusters in terms of design of CS courses. While in some learning design research there is evidence of common design practices [1, 90, 91] when comparing different disciplines, these preliminary findings seem to suggest three broad flavours of design in CS.

Future research should be carried out to identify and propose corresponding learning designs that would include Education 4.0 characteristics and thus transform CS courses. It is important for CS to follow trends in industry, while also providing future anticipation of possible changes. Furthermore, there is an urgent need to critically assess whether the concept of Education 4.0 is useful (or not) for CS. While some of the concepts of active learning, empowering students, and hands-on learning are increasingly common and well-supported by robust evidence, more research is needed to explore whether all elements of Education 4.0 are necessarily beneficial for learning or not.

A. Limitations and future research

There is an inherent systemic bias in terms of published outputs, as it is more likely that successful innovations and experiments are published than unsuccessful innovations, as well as "business as usual" approaches. Furthermore, with the rapid changes in CS and the shift in practice due to COVID-19, the reported findings might evolve over time. Another limitation is the search string that was used, whereby different key terms of search strings might have resulted in different outcomes. Nonetheless, using a robust 3 phase coding strategy we believe that we are the first to systematically review the pedagogical learning design decisions that CS educators make when designing innovative practice. By using the Education 4.0 characteristics our findings suggest three common flavors that CS educators use to design their practice. Future research should establish which of these common design practices work well for which groups of CS students, and for which specific knowledge, skills and competences. This will help to strengthen our evidence base and understanding of how to effectively design innovative CS courses that help to empower Education 4.0 in Industry 4.0.

VII. References

[1] K. Mangaroska, and M. N. Giannakos, "Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning," *IEEE Transactions on Learning Technologies*, vol. 12, no. 4, pp. 516-534, 2019.

[2] K. Mangaroska, B. Vesin, V. Kostakos *et al.*, "Architecting Analytics Across Multiple E-learning Systems to Enhance Learning Design," *IEEE Transactions on Learning Technologies*, pp. 1-1, 2021.

[3] M. Derntl, S. Neumann, D. Griffiths *et al.*, "The Conceptual Structure of IMS Learning Design Does Not Impede Its Use for Authoring," *IEEE Transactions on Learning Technologies*, vol. 5, no. 1, pp. 74-86, 2012.

[4] L. P. Macfadyen, L. Lockyer, and B. Rienties, "Learning Design and Learning Analytics: Snapshot 2020," *Journal of Learning Analytics*, vol. 7, no. 3, pp. 6-12, 12/17, 2020.

[5] B. Rienties, and L. Toetenel, "The impact of learning design on student behaviour, satisfaction and performance: a cross-institutional comparison across 151 modules," *Computers in Human Behavior*, vol. 60, pp. 333-341, 2016.

[6] K. P. Aničić, B. Divjak, and K. Arbanas, "Preparing ICT Graduates for Real-World Challenges: Results of a Meta-Analysis," *IEEE Transactions on Education*, vol. 60, no. 3, pp. 191-197, 2017.

[7] V. Garousi, G. Giray, E. Tüzün *et al.*, "Aligning software engineering education with industrial needs: A meta-analysis," *Journal of Systems and Software*, vol. 156, pp. 65-83, 2019/10/01/, 2019.

[8] L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching Practices of Software Testing in Programming Education," in 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1-9.

[9] European Commission/EACEA/Eurydice, *Teaching Careers in Europe: Access, Progression and Support*, Eurydice Report,, Luxembourg, 2018.

[10] Jisc. "Education 4.0 – transforming the future of education (through advanced technology)," 20 January 2021; https://www.youtube.com/watch?v=aVWHp8FsV1w.

[11] A. A. Hussin, "Education 4.0 made simple: Ideas for teaching," *International Journal of Education and Literacy Studies*, vol. 6, no. 3, pp. 92-98, 2018.

[12] G. Salmon, "May the Fourth Be with you: Creating Education 4.0," *Journal of Learning for Development*, vol. 6, no. 2, 07/18, 2019.

[13] R. Ferguson, C. Herodotou, F. Iniesto *et al.*, *Catalogue of new forms of teaching, learning and assessment in Computer Science in Edu 4.0 and related teachers' skills and competences*, University of Zagreb, Zagreb, 2021.

[14] Quality Assurance Agency, *Subject Benchmark Statement on Computing*, Quality Assurance Agency, Gloucester, UK, 2019.

[15] L. Toetenel, and B. Rienties, "Analysing 157 Learning Designs using Learning Analytic approaches as a means to evaluate the impact of pedagogical decision-making," *British Journal of Educational Technology*, vol. 47, no. 5, pp. 981–992, 2016.

[16] A. M. Harkins, "Leapfrog principles and practices: Core components of education 3.0 and 4.0," *Futures Research Quarterly*, vol. 24, no. 1, pp. 19-31, 2008.

[17] P. Fisk. "Education 4.0 … the future of learning will be dramatically different, in school and throughout life.," 20 January 2021; https://www.thegeniusworks.com/2017/01/future-education-young-everyone-taught-together/.

[18] C. A. Bonfield, M. Salter, A. Longmuir *et al.*, "Transformation or evolution?: Education 4.0, teaching and learning in the digital

age," *Higher Education Pedagogies,* vol. 5, no. 1, pp. 223-246, 2020/01/01, 2020.
[19] G. J. M. d. Costa, N. Silva, and T. Fonseca, "Moral reasoning in knowledge authoring: An e-learning 4.0 analysis!," *Computers in Education*, S. Abramovich, ed., pp. 135-154: Nova Science Publishers, 2012.
[20] V. Puncreobutr, "Education 4.0: New challenge of learning," *St. Theresa Journal of Humanities and Social Sciences,* vol. 2, no. 2, 2016.
[21] S. Suhaimi, "Education 4.0: The Impact of Computer Architecture and Organization Course on Students' Computer Anxiety and Computer Self-Efficacy," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 8, pp. 3022-3025, 12/15, 2019.
[22] T. Wallner, and G. Wagner, "Academic Education 4.0." pp. 155-159.
[23] M. Borge, Y. S. Ong, and S. Goggins, "A sociocultural approach to using social networking sites as learning tools," *Educational Technology Research and Development,* vol. 68, no. 3, pp. 1089-1120, 2020/06/01, 2020.
[24] P. Gestwicki, and B. McNely, "Interdisciplinary Projects in the Academic Studio," *ACM Trans. Comput. Educ.,* vol. 16, no. 2, pp. Article 8, 2016.
[25] C. L. Corritore, and B. Love, "Redesigning an Introductory Programming Course to Facilitate Effective Student Learning: A Case Study," *Journal of Information Technology Education: Innovations in Practice,* vol. 19, pp. 91-135, 2020.
[26] R. P. Medeiros, G. L. Ramalho, and T. P. Falcão, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Transactions on Education,* vol. 62, no. 2, pp. 77-90, 2019.
[27] H. W. Alomari, V. Ramasamy, J. D. Kiper *et al.*, "A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education," *Heliyon,* vol. 6, no. 5, pp. e03917, 2020/05/01/, 2020.
[28] D. Moher, A. Liberati, J. Tetzlaff *et al.*, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *Int J Surg,* vol. 8, no. 5, pp. 336-341, 2010.
[29] R. J. House, P. J. Hanges, M. Javidan *et al.*, *Culture, leadership and organizations: The GLOBE study of 62 Societies,* Thousand Oaks, CA: Sage Publications Inc, 2004.
[30] U. Schäfer, "Teaching Modern C++ with Flipped Classroom and Enjoyable IoT Hardware," in 2019 IEEE Global Engineering Education Conference (EDUCON), 2019, pp. 910-919.
[31] M. Apiola, E. Lokkila, and M.-J. Laakso, "Digital learning approaches in an intermediate-level computer science course," *The International Journal of Information and Learning Technology,* vol. 36, no. 5, pp. 467-484, 2019.
[32] A. C. Burrows, and M. Borowczak, "Computer science and engineering: utilizing action research and lesson study," *Educational Action Research,* vol. 27, no. 4, pp. 631-646, 2019/08/08, 2019.
[33] P. Degener, V. Haak, C. Gold-Veerkamp *et al.*, "Towards the Vision of an LMS Integrated, Browser-Based Simulation to Program LEGO MindStorms EV3s in ANSI-C," in 2019 IEEE Global Engineering Education Conference (EDUCON), 2019, pp. 89-94.
[34] P. E. Dickson, T. Dragon, and A. Lee, "Using Undergraduate Teaching Assistants in Small Classes," in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, Washington, USA, 2017, pp. 165–170.
[35] P. Dondio, and S. Shaheen, "Is StackOverflow an Effective Complement to Gaining Practical Knowledge Compared to Traditional Computer Science Learning?," in Proceedings of the 2019 11th International Conference on Education Technology and Computers, Amsterdam, Netherlands, 2019, pp. 132–138.
[36] W. Fisher, C. Rader, and T. Camp, "Online programming tutors or paper study guides?," in 2016 IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-6.
[37] T. Frevert, A. Rorrer, D. J. Davis *et al.*, "Sustainable Educational Innovation Through Engaged Pedagogy and Organizational Change," in 2018 IEEE Frontiers in Education Conference (FIE), 2018, pp. 1-5.
[38] N. Giacaman, and G. De Ruvo, "Bridging Theory and Practice in Programming Lectures With Active Classroom Programmer," *IEEE Transactions on Education,* vol. 61, no. 3, pp. 177-186, 2018.
[39] H. Hosseini, M. Hartt, and M. Mostafapour, "Learning IS Child's Play: Game-Based Learning in Computer Science Education," *ACM Trans. Comput. Educ.,* vol. 19, no. 3, pp. Article 22, 2019.
[40] J. A. Parejo, J. Troya, S. Segura *et al.*, "Flipping Laboratory Sessions: An Experience in Computer Science," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje,* vol. 15, no. 3, pp. 183-191, 2020.
[41] H. Park, and Y. Kim, "CLIK: Cloud-based Linux kernel practice environment and judgment system," *Computer Applications in Engineering Education,* vol. 28, no. 5, pp. 1137-1153, 2020.
[42] C. Pilkington, "Questioning the Value of Vodcasts in a Distance Learning Theoretical Computer Science Course," Cham, 2017, pp. 83-98.
[43] N. Shi, Z. Min, and P. Zhang, "Effects of visualizing roles of variables with animation and IDE in novice program construction," *Telematics and Informatics,* vol. 34, no. 5, pp. 743-754, 2017/08/01/, 2017.
[44] W. Silva, I. Steinmacher, and T. Conte, "Students' and Instructors' Perceptions of Five Different Active Learning Strategies Used to Teach Software Modeling," *IEEE Access,* vol. 7, pp. 184063-184077, 2019.
[45] B. Tyler, and M. Abdrakhmanova, "Flipping the CS1 and CS2 classrooms in Central Asia," in 2016 IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-5.
[46] R. Caceffo, G. Gama, and R. Azevedo, "Exploring Active Learning Approaches to Computer Science Classes," in Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA, 2018, pp. 922–927.
[47] N. Aghaee, and C. Keller, "ICT-supported peer interaction among learners in Bachelor's and Master's thesis courses," *Computers & Education,* vol. 94, pp. 276-297, 2016/03/01/, 2016.
[48] N. Alasbali, and B. Benatallah, "Open source as an innovative approach in computer science education A systematic review of advantages and challenges," in 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), 2015, pp. 278-283.
[49] F. Alegre, J. Moreno, T. Dawson *et al.*, "Computational Thinking for STEM Teacher Leadership Training at Louisiana State University," in 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), 2020, pp. 1-2.
[50] B. Berikan, and S. Özdemir, "Investigating "Problem-Solving With Datasets" as an Implementation of Computational Thinking: A Literature Review," *Journal of Educational Computing Research,* vol. 58, no. 2, pp. 502-534, 2020.
[51] A. R. Bielefeldt, M. Polmear, C. Swan *et al.*, "An overview of the microethics and macroethics education of computing students in the United States," in 2017 IEEE Frontiers in Education Conference (FIE), 2017, pp. 1-9.
[52] M. Borowczak, and A. C. Burrows, "Interactive Web Notebooks Using the Cloud to Enable CS in K-16+ Classrooms and PDs," in 2017 ASEE Annual Conference & Exposition, 2017.
[53] A. C. Burrows, and M. Borowczak, "Hardening freshman engineering student soft skills," in Session W1A First Year Engineering Experience (FYEE) Conference, 2017, pp. 1-5.
[54] N. A. Bushmeleva, and T. A. Baklashova, "Methodological teaching system of mathematical foundations of formal languages as a means of fundamentalization of education," *EURASIA Journal of Mathematics, Science and Technology Education,* vol. 13, no. 8, pp. 5141-5155, 2017.
[55] G. Carrascal, A. A. del Barrio, and G. Botella, "First experiences of teaching quantum computing," *The Journal of Supercomputing,* vol. 77, no. 3, pp. 2770-2799, 2021/03/01, 2021.
[56] M. J. Casañ, M. Alier, and A. Llorens, "Teaching Ethics and Sustainability to Informatics Engineering Students, An Almost 30 Years' Experience," *Sustainability,* vol. 12, no. 14, pp. 5499, 2020.
[57] J. Chamberlin, J. Hussey, B. Klimkowski *et al.*, "The Impact of Virtualized Technology on Undergraduate Computer Networking Education," in Proceedings of the 18th Annual Conference on

Information Technology Education, Rochester, New York, USA, 2017, pp. 109–114.

[58] M. Cobos, and S. Roger, “SART3D: A MATLAB toolbox for spatial audio and signal processing education,” *Computer Applications in Engineering Education*, vol. 27, no. 4, pp. 971-985, 2019.

[59] F. Fagerholm, A. Hellas, M. Luukkainen *et al.*, “Designing and implementing an environment for software start-up education: Patterns and anti-patterns,” *Journal of Systems and Software*, vol. 146, pp. 1-13, 2018/12/01/, 2018.

[60] E. Juárez, R. Aldeco-Pérez, and J. M. Velázquez, “Academic approach to transform organisations: one engineer at a time,” *IET Software*, vol. 14, no. 2, pp. 106-114, 2020.

[61] M. C. Lewis, and L. L. Lacher, “Teaching Modern Multithreading in CS2 with Actors,” in 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2020, pp. 292-299.

[62] Z. Liang, and M. A. Chapa-Martell, “A Top-Down Approach to Teaching Web Development in the Cloud,” in 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2018, pp. 32-39.

[63] A. Llorens, J. Berbegal-Mirabent, and X. Llinas-Audet, “Aligning professional skills and active learning methods: an application for information and communications technology engineering,” *European journal of engineering education*, pp. 1-14, 2016.

[64] E. Mäkiö, E. Yablochnikov, A. W. Colombo *et al.*, “Applying Task-centric Holistic Teaching Approach in Education of Industrial Cyber Physical Systems,” in 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), 2020, pp. 359-364.

[65] B. S. Santos, J.-M. Dischler, V. Adzhiev *et al.*, “Distinctive Approaches to Computer Graphics Education,” *Computer Graphics Forum*, vol. 37, no. 1, pp. 403-412, 2018.

[66] M. Seyam, and D. S. McCrickard, “Teaching Mobile Development with Pair Programming,” in Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA, 2016, pp. 96–101.

[67] S. Alsaif, A. S. Li, B. Soh *et al.*, “The Efficacy of Facebook in Teaching and Learning: Studied via Content Analysis of Web Log Data,” *Procedia Computer Science*, vol. 161, pp. 493-501, 2019/01/01/, 2019.

[68] K. A. Behnke, B. A. Kos, and J. K. Bennett, “Computer Science Principles: Impacting Student Motivation & Learning Within and Beyond the Classroom,” in Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, VIC, Australia, 2016, pp. 171–180.

[69] J. Broisin, R. Venant, and P. Vidal, “Lab4CE: a remote laboratory for computer education,” *International Journal of Artificial Intelligence in Education*, vol. 27, no. 1, pp. 154-180, 2017.

[70] K. Buffardi, and P. Valdivia, “Bug Hide-and-Seek: An Educational Game for Investigating Verification Accuracy in Software Tests,” in 2018 IEEE Frontiers in Education Conference (FIE), 2018, pp. 1-8.

[71] P. Charlton, and K. Avramides, “Knowledge Construction in Computer Science and Engineering when Learning Through Making,” *IEEE Transactions on Learning Technologies*, vol. 9, no. 4, pp. 379-390, 2016.

[72] R. Q. Gonçalves, C. A. G. von Wangenheim, J. C. R. Hauck *et al.*, “An Instructional Feedback Technique for Teaching Project Management Tools Aligned With PMBOK,” *IEEE Transactions on Education*, vol. 61, no. 2, pp. 143-150, 2018.

[73] C. Goumopoulos, P. Nicopolitidis, D. Gavalas *et al.*, “A distance learning curriculum on pervasive computing,” *International Journal of Continuing Engineering Education and Life Long Learning*, vol. 27, no. 1-2, pp. 122-146, 2017.

[74] M. Knobelsdorf, C. Frede, S. Böhne *et al.*, “Theorem Provers as a Learning Tool in Theory of Computation,” in Proceedings of the 2017 ACM Conference on International Computing Education Research, Tacoma, Washington, USA, 2017, pp. 83–92.

[75] R. I. Munkvold, “Game lab: a practical learning approach for game development,” in European Conference on Games Based Learning, 2017, pp. 472-479.

[76] L. N. Paschoal, B. R. N. Oliveira, E. Y. Nakagawa *et al.*, “Can we use the Flipped Classroom Model to teach Black-box Testing to Computer Students?,” in Proceedings of the XVIII Brazilian Symposium on Software Quality, Fortaleza, Brazil, 2019, pp. 158–167.

[77] D. Pawelczak, "Comparison of traditional lecture and flipped classroom for teaching programming." pp. 391-398.

[78] C. Peng, “Introductory Game Development Course: A Mix of Programming and Art,” in 2015 International Conference on Computational Science and Computational Intelligence (CSCI), 2015, pp. 271-276.

[79] M. S. Peteranetz, A. E. Flanigan, D. F. Shell *et al.*, “Computational Creativity Exercises: An Avenue for Promoting Learning in Computer Science,” *IEEE Transactions on Education*, vol. 60, no. 4, pp. 305-313, 2017.

[80] I. Pivkina, “Peer learning assistants in undergraduate computer science courses,” in 2016 IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-4.

[81] J. Ruiz, E. Serral Asensio, and M. Snoeck, “Learning UI Functional Design Principles Through Simulation With Feedback,” *IEEE Transactions on Learning Technologies*, vol. 13, no. 4, pp. 833-846, 2020.

[82] F. A. Salem, I. W. Damaj, L. Hamandi *et al.*, “Effective Assessment of Computer Science Capstone Projects and Student Outcomes,” *iJEP*, vol. 10, no. 2, pp. 72-93, 2020.

[83] M. Seyam, D. S. McCrickard, S. Niu *et al.*, “Teaching mobile application development through lectures, interactive tutorials, and Pair Programming,” in 2016 IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-9.

[84] P. A. Silva, B. J. Polo, and M. E. Crosby, "Adapting the Studio Based Learning Methodology to Computer Science Education," *New Directions for Computing Education: Embedding Computing Across Disciplines*, S. B. Fee, A. M. Holland-Minkley and T. E. Lombardi, eds., pp. 119-142, Cham: Springer International Publishing, 2017.

[85] R. Tanaka, R. Ferreira da Silva, and H. Casanova, “Teaching Parallel and Distributed Computing Concepts in Simulation with WRENCH,” in 2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC), 2019, pp. 1-9.

[86] A. Tlili, F. Essalmi, M. Jemni *et al.*, "Towards Applying Keller’s ARCS Model and Learning by doing strategy in Classroom Courses," *Innovations in Smart Learning*. pp. 189-198.

[87] C. Troussas, A. Krouska, and C. Sgouropoulou, “Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education,” *Computers & Education*, vol. 144, pp. 103698, 2020/01/01/, 2020.

[88] D. Winiecki, and N. Salzman, “Teaching Professional Morality & Ethics to Undergraduate Computer Science Students through Cognitive Apprenticeships & Case Studies: Experiences in CS-HU 130 ‘Foundational Values’,” in 2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), 2019, pp. 1-3.

[89] Z. J. Wood, J. Clements, Z. Peterson *et al.*, “Mixed Approaches to CS0: Exploring Topic and Pedagogy Variance after Six Years of CS0,” in Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA, 2018, pp. 20–25.

[90] B. Rienties, W. Kaper, K. Struyven *et al.*, “A review of the role of Information Communication Technology and course design in transitional education practices,” *Interactive Learning Environments*, vol. 20, no. 6, pp. 563-581, 2012.

[91] B. Rienties, L. Toetenel, and A. Bryan, "“Scaling up” learning design: impact of learning design activities on LMS behavior and performance," *5th Learning Analytics Knowledge conference*, pp. 315-319, New York: ACM, 2015.

Bart Rienties received his PhD at Maastricht University in the Netherlands in 2010. From 2010-2013 he worked as (senior) lecturer at University of Surrey, and since 2014 has worked at the Institute of Educational Technology at the Open University UK. He has been a Professor of Learning Analytics since 2017 and has published over 250 articles and conference proceedings on learning analytics, learning design, computer-supported collaborative learning, internationalization, and professional development.

Rebecca Ferguson received her PhD from The Open University in the UK in 2010. Since then, she has worked at The Open University and was appointed as Professor of Learning Futures in 2020. Her publications include Educational Visions: The Lessons from 40 Years of Innovation (London 2019) and Augmented Education (New York 2014). Her research focuses on innovations in pedagogy, online learning, learning at scale and learning analytics.

Dalibor Gonda obtained a PhD. degree at the University of Constantine the Philosopher in Nitra, Slovakia. Since 2016, he has been working as an assistant at the University of Žilina in Žilina, where he lectures on statistics and algebra. He researches various aspects of teaching mathematics, statistical data processing and the use of spatial statistics in location tasks.

Goran Hajdin received his PhD at the University of Zagreb in Croatia in 2014. From 2007 he works at Faculty of Organization and Informatics, University of Zagreb on the courses related to pedagogy, didactics and methods of teaching informatics. He is a head of lifelong learning module for teacher education since 2017. He publishes research on topics related to pedagogy, educational technology and teaching methods with special focus on informatics.

Christothea Herodotou received her PhD from the Institute of Education, UCL in 2009. She worked as a primary teacher for 7 years and since 2013 as a postdoctoral researcher at the Open University UK. In 2018, she became an Associate Professor. She is now leading two award-winning university wide initiatives about community citizen science (nQuire) and predictive learning analytics (OUAnalyse).

Francisco Iniesto received his PhD at The Open University in the UK in 2020. His background is as a Computer Engineer with extensive experience in IT consulting and software development. From 2018 he works as a Research Associate and Associate Lecturer at the Institute of Educational Technology at the Open University. His areas of research and publications are in inclusive design, accessible educational technology, and open education.

Ariadna Llorens Garcia (Member, IEEE) is an associate lecturer in the Management Department at the Universitat Politècnica de Catalunya – Barcelona Tech (UPC). She is an Industrial Engineer and holds a PhD in Management from UPC and a MBA in Administration from ESADE. Her research interests are in the fields of engineering education, university-business cooperation and innovative learning methodologies.

Henri Muccini has a Ph.D. in Computer Science and is a Professor at the University of L'Aquila, Italy. His research interests are in the Software Engineering field, and specifically on software architecture descriptions and analysis, machine learning for architecting quality systems, software migration to microservices, and model-driven software architectures.

Julia Sargent received her PhD at Loughborough University in the UK in 2018. From 2018 she worked as a Research Fellow at the Open University and since 2019 has worked as a Lecturer at the Institute of Educational Technology at the Open University UK. She publishes research on aspects such as educational technology, physical education and pedagogy.

Sirje Virkus received her PhD at Manchester Metropolitan University in the UK in 2011. From 1985 she has worked at Tallinn University as a lecturer and since 2012 as a Professor of Information Science. She is the Head of the Study Area of Information Sciences at the School of Digital Technologies of Tallinn University. Her research interests are focused on the development of information-related competencies, ICT innovation in education and internationalization. She has written more than 200 research publications.

Maria Vittoria Isidori received her cognitive neuroscience PhD at the Padova/L'Aquila University in Italy in 1998. From 2008 she works at Department of Human Studies University of L'Aquila on the courses related to pedagogy, didactics and methods of inclusive teaching. She is a head of lifelong learning module for teacher education since 2014. She publishes research on topics related to pedagogy, learning process, inclusive didactic, educational and teaching methods with special focus on Special Need SEN.