The Global Challenge of Designing E-learning Tools for Computational Thinking: A Comparison Between East Asia and Scandinavia

Kasper Kristensen[1], Emanuela Marchetti[2], and Andrea Valente[1](✉)

[1] The Maersk Mc-Kinney Moller Institute, SDU Game Development and Learning Technology, Odense, Denmark
{kakr,anva}@mmmi.sdu.dk
[2] Media, Department for the Study of Culture, University of Southern Denmark, Odense, Denmark

Abstract. In this paper we investigate status and trends in the pedagogy of Computational Thinking (CT), in Scandinavia and Eastern Asian countries. A more detailed comparison is drawn between two specific countries: Denmark and Taiwan. Combining a literature review on official information about the implementation of this new subject in schools, interviews with experts and practitioners, we identify core aspects in the pedagogy of CT across sociocultural differences, such as: the role and relation between formal and non-formal learning, the relation between CT and other school subjects, coding as an unavoidable part of CT as a subject, the tendency to adopt and adapt globally shared materials originally imported from the North American educational discourse. We also noticed that in Danish primary and secondary schools, current orchestration strategies in CT-related learning activities tend to leverage hands-on tinkering, peer-learning, and collaborative/group-based problem-solving; similar strategies are adopted in Taiwanese clubs. In this respect, we identify a lack of support for group work in existing e-learning tools for coding. Our main contribution is the definition of a scenario and requirements for a new class of e-learning tools, capable of supporting group-based CT learning activities across different culture. We are currently organizing a series of observations of the teaching practices of coding within CT, in cooperation with our network of contacts in Taiwan and Japan. Future work involves the development of a prototype of the new e-learning tool, iteratively, involving experts from Scandinavia and East Asia.

Keywords: Computational thinking · Scandinavia and East Asia · Tools · Programming · Cooperation

P. Zaphiris and A. Ioannou (Eds.): HCII 2021, LNCS 12784, pp. 469–487, 2021.
https://doi.org/10.1007/978-3-030-77889-7_33

1 Introduction

Computational Thinking (CT) [10] has recently seen a nearly universal recognition globally [3] and school systems across the world are at various stages of implementing new curricula including digital literacy, ICT and CT as core subjects. Our study has the double goal of gaining an understanding of the emerging pedagogy of CT as a new school subject from a global perspective, and what are the consequences for the development of e-learning tools that could be applicable across different countries and pedagogical practices.

We start by looking at existing definitions of CT, to appreciate the complexity of finding a common, globally shared, practical definition; this problem stretches from primary schools to higher education. We proceed with a preliminary case study on a comparison between East Asia and Scandinavia, focusing on Denmark and Taiwan, as examples of different pedagogical practices potentially embodying different sociocultural values and needs. The comparison is based on literature reviews [3, 4], interviews [2], and dialogue with local researchers. Our interests for these specific countries emerged from noticing overlapping sociocultural factors, such as: relatively comparable (i.e. small) size of the country, advanced digitization of their society, both committed to implementing CT in their school curricula, and both influenced by North-American advancements in CT and in digital innovation. Our preliminary findings show similarities between the approaches to CT of the two countries, in relation to the importance of the social aspects of learning and the relevance of coding as a central, hands-on activity in CT practices. We are aware of the distinction between coding and programming [9], and we agree that *coding* is the activity of converting instructions from human form to a code runnable by a machine, while *programming* is often taken to include a larger scope and involves also other activities in software development (e.g. planning, debugging, testing, deployment, etc.). Coding is therefore part of programming, and programming is considered an important 21st century skill, providing an advantage in the job market to technical and humanistic graduates [11]; however, programming is not yet a primary school subject, not in Scandinavia nor in East Asia. Furthermore, in the discussion about the nature and definition of CT, it is typically agreed that CT should not be considered simply a programming curriculum for pupils. So we propose to consider coding as a challenging but unavoidable and highly desirable part of CT. We decided, therefore, to analyze existing e-learning tools for coding, targeted at our target group, and look at them through the lens of learning theories such as constructivism, Vygotsky's *Zone of Proximal Development* [22], and peer learning more in general. We found that existing IDE for young beginners are designed with a single user in mind: this might not support adequately current the orchestration strategies for CT that we have found in our case study, which typically leverage forms of peer-learning and collaborative problem-solving. The main contributions of this paper are, therefore, the identification of a lack of support for group work in existing e-learning tools for coding, targeted at primary schools; and the definition of a scenario and requirements for a new class of e-learning tools, capable of supporting group-based CT learning activities across different culture.

The rest of the paper discusses the complexity of finding a definition for CT (Sect. 2) and the theoretical work we used to look at CT in this study; Sect. 3 introduces our case study and our findings, that are further discussed in Sect. 4. Section 5 presents guidelines and specifications for the design of e-learning tools that better fit the emergent needs of the pedagogy of CT. Section 6 concludes the paper.

2 CT – A Concept in Search of a Definition

CT has seen widespread and ongoing adoption both in Europe [5] and East Asia (Kristensen 2020), hence their school systems are in various stages of implementing CT into their school curriculums, focusing more specifically on the cases of Denmark and Taiwan.

Our comparison among countries builds on a conceptual framework, informed by existing complementary views of CT. The term CT was originally coined by Papert in his book *Mindstorms* [2, 30] 30 years ago, but the definition and focus have evolved over the decades. Papert [2, 30] originally envisioned the personal computer as a great facilitator for assimilating knowledge of any kind, as the personal computer's adaptive nature would allow it to be tailored to any individual interest, functioning as a catalyst for intrinsic motivation. Papert laments the fact that computers were yet not powerful enough to create fully engaging activities, but stating than when the technology is sufficiently mature, it will allow CT to be integrated into everyday life [2]. Papert's definition of CT was to be able to use computers for everyday tasks, which could sound already surpassed given the average computer skills of today's pupils and students.

Current CT is clearly not yet mature and each country is trying to figure out how a person becomes a computational thinker, and to cope with the lack of a widely agreed-upon definition of CT, in this paper we follow Wing [10] and define CT as the ability to solve problems *algorithmically* (i.e. in such a way that a computer could be used to solve them). In this sense CT can be seen also as a problem solving method, supported by computers and algorithms. Therefore, CT learners would need to understand how data and information is stored and processed within a computer. A main challenge in the quest for turning CT into a school subject is then to identify which set of skills and knowledge should be possessed by citizens in the 21st century [4, 10, 29], to understand the impact of digital and pervasive systems on society and their individual rights, while being able to use the same systems in their daily jobs, without being IT professionals. As automatic systems become more ubiquitous, the ability to understand and use those systems will be paramount to become an effective agent in the present and future digital world, not unlike reading and writing skills are today, as exemplified by the popularization of the term *digital literacy* [29].

In our comparative study, we coped with this challenge referring to a simple categorization of CT chore elements, designed and widely used for teaching CT notably in the USA, England and Taiwan [5], which gives a general overview of the different elements of CT, such as: decomposition, pattern recognition, abstraction, and algorithms. Decomposition is the first step in the process of dividing a problem into discrete parts. Each part can then be tackled individually, flattening the complexity of the overall problem. Pattern Recognition is the second step, in which commonalities are identified, which could be repeating sequences, generalized categories or other similarities observed in the problem. Abstraction is the third step, in which the solutions are modeled and abstracted. A good CT solution should be general, solving multiple problems of a similar nature and not merely the initial specific problem. The last step is represented by algorithms, in which the problem is expressed as a series of steps necessary to find the solution to the problem.

In conclusion, a universally accepted definition of CT is still missing, and each country is adopting a customized blending of elements CT, referring to different international and national sources [6]. Moreover, the practical implementation of CT in primary school requires to consider more aspects than simply the core elements of CT, such as: classroom orchestration, support for teachers' competences, and development of suitable textbook and online materials in the appropriate language.

2.1 Theoretical Framework

Since we are interested in the learning and pedagogy of CT in different countries, we need to formulate a pedagogical lens to analyze and compare CT approaches. In this sub-section we outline the theories that we adopt in our analysis.

Our study builds mainly on Vygotsky's learning theory [22], which constitutes the basis of constructivism, the approach also adopted by Papert [30]. Instead of focusing on the transmission of knowledge from teacher to student, constructivism is concerned with the process of the learner of gaining knowledge by exploring and applying knowledge in a practical context acting with material toys or tools. As such constructivism sees meaning making as a qualitative, exploratory process [8], in which the learner actively forms new connections with his or her own existing knowledge and understanding, forming new and sharper understanding.

In Vygotsky's theory [22], learning is seen as an inquiry-based process of knowledge acquisition, instead of a one-way street, in which the teacher with eloquent and succinct languages transfers knowledge from himself to his students. Learners are seen in Vygotsky as independently engaging in their learning, participating in social activities with peers and adults, who might provide support when the learners reach their Z*one of Proximal Development*, defined as a natural boundary between what the leaners can accomplish on his or her own, and what which he or she can understand with proper facilitation [22]. The Zone of Proximal Development is delimited by preexisting knowledge of the learners and by the cognitive developments of young learners, in relation to what they can understand for their age. In this respect, Vygotsky's theory offered already a counterpoint to traditional learning theories, proposing that the student actively builds meaning through a interplay with the teacher, leveraging his or her own preexisting understanding as a foundation.

Vygotsky generally assumes that learners acquire knowledge through a dialectic collaboration with their teachers [22]. However, this dialectic collaboration can be extended to more experienced peers, hence leading to forms of peer learning. This scenario is explored in theories of *apprenticeship learning* [23, 29, 31], which are founded on Vygotsky's theory and build on observations from craftsman apprenticeships. In apprenticeship learners are seen as engaging with peers and adult supervisors in different social activities such as food preparation, which involve a goal not strictly related to learning, that would be the creation of a dish and the social enjoyment of it. These activities involve also a set of material objects and tools, and the adult are in charge in segmenting the activity at hand in smaller tasks with related goals, which could be conducted step by step by the learners under supervision [23, 31]. When encountering difficulties either adults or more expert peers will provide the needed support. As the novice's skills improve the experts will gradually remove themself from the process, gradually enabling the apprentice to complete the activity on his or her own [23, 31]. As such apprenticeship learning includes the principle of *scaffolding* [31], where support from the experts is slowly removed and the overall learning goal is to enable the apprentice to perform the activity independently in future occasions [23]. From the perspective of the Zone of Proximal Development, scaffolding aims to move further the ability of the learners from what they can do with help, to full control on the activity.

Apprenticeship learning highlights the importance of the physical learning environment [23, 31], as resources for meaning making. At its core apprenticeship is learning through practical application of skills and knowledge, in which peer-learning is seen students have an ideal environment for questioning, evaluating and restructuring existing knowledge [23]. This transformation of knowledge can happen as a result of discussion between individuals explaining, or defending their own understanding and viewpoints. In order for peer-learning to be effective, it can be argued that one of the students should act as the expert, being slightly ahead of the other students but both will learn and develop through the process [32].

Forms of peer learning are targeted by *playful learning* and *gamification*, according to which learning can take place through facilitated play. Playful learning is seen as fostering motivation, social engagement, and deeper forms of understanding in the learners [13, 22]. According to Vygotsky, play enables children to develop superior cognitive abilities, enabling them to go beyond their present reality, reflecting on the consequence of hypothetical courses of action. In so doing, children start early to practice forms of conceptual thinking [22]. Gee [13] argues that players create their own *affinity spaces,* seen as imaginary realities defined by the players actions, mutual interests, and the surrounding context.

Similar dynamics take place in the adoption of gamification, in which certain structures from games are applied to learning activities, so to create an engaging motivational reward structure to foster the learners' motivation [14]. Gamification has been applied to learning, but also teambuilding and innovation practice in professional contexts, healthcare, and physical activities. Gamification takes inspiration from reward structures developed from social competitive games, computer games and sports. Therefore, central elements of gamification include social engagement, daily bonuses, and streaks, a typical example would be virtual badges collected through a series of quests while playing a computer game or a public leader-board showing the accomplishment of all participants [14].

According to our preliminary data, forms of playful learning and gamification are being explored to create CT learning activities for primary and secondary schools, and especially in the case of clubs.

3 Case Study: CT Across Countries

This section describes the current status of the discussions and implementations of CT as a primary and secondary school subject, in Scandinavia and East Asia.

3.1 State of CT in Scandinavia and Denmark

In Scandinavia all countries tend to work in close cooperation when global challenges arise, and this is happening such as the global challenge of turning CT into a school subject. In a commissioned report [5] about the implementation of CT in Scandinavia, it is stated that Denmark, Sweden, Norway and Finland are all pursuing this complex task, in similar ways, even if they are at different stages of implementation. More specifically three approaches are identified in Scandinavia to the inclusion of CT in the school curriculum: *"a cross-curriculum strategy, accommodation in subject(s) already being taught, establishment of a new, purposely-designed subject. […] Finland and Sweden have adopted a blend of cross-curriculum and single subject integration, where the strongest subject link - in terms of coverage and learning outcomes - appears to be with mathematics."*. Instead Denmark and Norway adopted *"pilot initiatives at lower secondary level"* and they appear to *"position CT within an elective subject that has strong links to computer science, leveraging […] learning contents for developing CT skills."*. Teachers' training is also a central focus in all Scandinavian countries, which have a tradition of defining frameworks for school subjects, but leaving the teachers freedom in the micro-level management of contents in the actual courses.

In Denmark CT has been implemented from the top, it has been pushed from a political institutional level, starting recently in high school with a new course called "Informatik" by the Ministry of Education [17]. Afterwards, CT experiment was moved down to elementary school, where a large trial is currently being conducted since 2019 and will finish in summer 2021. The aim of this trial is to test the inclusion of CT in Danish schools as an independent course or within other courses. The new primary school CT course is named "Teknologiforståelse", which translates to "Technology Understanding" [18]. The core competences of the new course (translated from the official Danish documentation) include:

- *Digital empowerment* - critical investigation and understanding of digital artefacts like apps,
- *Digital Design* - with focus on the design processes,
- *Computational Thinking* - where pupils must analyze, model, and structure data and data processing,
- *Technological skills* - which include mastering computer systems, digital tools, and programming.

So defined, the Danish school system considers programming, and not only coding, as a skill to be included in Computational Thinking.

The implementation of the "Informatik" course is an interesting case: since it is a more mature subject than "Teknologiforståelse", but it represents a close relative and a kind of precursor, a large amount of data is available about the deployment and challenges of "Informatik". Compared to other courses within the Danish school system "Informatik" is unique, as it specifies which pedagogical approaches should be used in teaching its subject matter, not to the exclusion of possible other approaches, but it is worth noting that in Danish education, such decisions are normally left to the teachers' discretion. Similarly, the choice of programming languages, software, or other tools, to be used within the course, are also left to individual teachers, this represents a more typical balance of responsibilities in the Danish school system. Hence, Danish teachers are currently experiencing additional pressure than usual in experimenting with CT, as they are given more rigid instructions from the political level to teach an undefined subject, while not being experts themselves. In this respect, an interesting challenge identified during the ongoing trial with CT is the competence and knowledge gap between teachers and those who are developing the CT curriculum [27]. In Denmark, a curriculum is usually defined in the form of lesson plans and assignments, which teachers and schools can choose from, as a supplement to their own lessons and assignments. However, in the case of "Informatik", the curriculum is reportedly formulated in a specialized IT language, which makes it hard for the teachers to understand how to use and adapt the provided lessons plan to their needs.

Recent lockdowns of schools due to the Covid-19 pandemic has caused additional challenges, as high school students and pupils cannot be followed as closely as usual. Moreover, a reprioritization of subjects has led to focus on core subjects, such as Danish, Mathematics, and foreign languages, with the penalization of lab-based subjects, like sciences, art and design, and CT.

Besides schools, in Denmark CT is being introduced also through various types of computer science *clubs,* where children can experiment with microcontrollers, programming, 3d printers, and digital art after their classes. These clubs are volunteer organizations financed by the municipality, which have become very spread in Denmark over the last couple of years. These clubs offer a variety of informal learning activities in CT related topics, taught by expert volunteers from local education institutions or from within the IT field. Hence, these clubs have often a strong profile and can offer relevant insights for developing a CT curriculum.

Both in Danish schools and clubs, two main pedagogical approaches are used in teaching the subject "Informatik" and in the "Teknologiforståelse" trial such as: *Worked Examples* and *Use-Modify-Create*. In the worked example approach, a solution is to show a solution to a problem, and to show the thought process behind its implementation (as in [19] and [20]). Worked Examples are often produced with multiple modalities in mind such as audio and visual, videos are extensively used as these allow pupils to engage with the example at their own pace. Worked Examples demystify the problem solving process, by showing the process itself in a step-by-step fashion, and considerations of what goes into a design, which is not possible to show in finished code or completed problem solutions, hence Worked Examples are especially useful at early stages of skill acquisition, guiding the pupils step by step showing theory and relevant examples saliently shown in the material [19].

The other pedagogical approach found in "Informatik" is Use-Modify-Create, which was proposed by Lee et al. [21] as a promising pattern for teaching CT. In this pedagogical approach we allow the students to first engage with a given product, for instance a digital game or other interactive media, and afterwards to modify its code.

This process could be informed by improvements the students wish to implement, after having used the product and having noticed small bugs in the code. As the students' understanding of the code increases, they can move from the modification phase to the creation phase: first by implementing new features to the existing product and later to create completely new products (as also suggested in the document from the Danish Ministry of Education [17]).

CT experimentation in Denmark is taking place across educational contexts, moreover, more data are available from high schools and the “Informatik” subject. However, because of fundamental differences among high schools, primary schools and clubs, a pedagogical approach which is highly successful in one setting might still need adapted to another or might need to be discarded entirely. Therefore, it is difficult to directly base a new CT course for elementary schools merely on the pedagogical patterns that has worked for “Informatik” or in the clubs, which are typically informal, participatory, and social. In conclusion, the current Danish and Norwegian trial is providing data and observations that are needed to shed light on these issues. Interestingly we are seeing that hands-on methods like Worked Examples and Use-Modify-Create seem to emerge as widely adopted pedagogical approaches, in Denmark and possibly in other countries attempting the same implementation of CT in schools.

3.2 State of CT in East Asian Countries and Taiwan

In order to gain insight into the development of CT in East Asia, a literature review was conducted. The goal of the literature review was to show the current academic discourse around CT for each country of interest [4]. The focus was on smaller East Asian countries, because they have more similarities with Denmark and the other Scandinavian countries: they are of rather comparable in size, they are striving to implement CT following the North American effort, and they typically have a good level of digitization of the society. As such we assume that they would be frontrunners in the implementation of CT in schools, and that their approaches and contingencies would be more comparable to the Danish context, as opposed to larger countries like China [4].

This review was compiled following a *systematic literature review* method, hence a series of papers was selected and analyzed from prominent conferences within the field of education with a clear focus on CT. We chose conferences held within the Asian region, as these were more likely to include local studies than similar conferences held overseas (a more complete list of criteria is presented in [4]). Table 1 shows a summary of the data found in our review, based on our categorization and thematic analysis of the papers.

The adoption of CT for the East Asian region is developing at a similar pace as in Europe [5]. No government in the East Asian region is completely disregarding the importance of information technology and we see an increasing focus on CT in the area: even if not all country plans to directly teach CT or programming, they are all working towards integrating digital tools in schools.

From our survey, the countries that seem to be of most interest for our study are Hong Kong, South Korea, and Taiwan. In Taiwan CT has been already included into the curriculum for grade 7 through 12; CT is taught for 1 h a week for the first two grades and 2 h for the remaining 3 grades, initially focusing on coding, hence covering control structures, and data structures such as arrays and trees [15]. The official textbook that is going to be used for CT in Taiwanese primary schools suggests the use of MIT's Scratch for the initial courses and AppInventor for more advanced grades. Adopting MIT tools offers many advantages, they are high-quality and well-tested, web-bases and freely accessible, moreover, it is easy to find many books based on the use of such tools for both young adults and children. Hence, following MIT's and North American research appears to be regional as well as global trend.

Table 1. Summary of findings from our literary review.

Country	Status of CT Implementation	Focus/details
China	Implemented	Implemented a new national curriculum in 2017, which included CT
Japan	Working towards	Programming as a compulsory subject for students in primary and secondary schools and should be fully implemented by 2022
Hong Kong	Implemented	ICT and CT has been integrated into the curriculum since 2017, and is currently finalized a supplementary curriculum specifically for CT
Macau	No plans	Focused on computer literacy and the usage of digital tools such as word processing, spreadsheets, and database applications
Mongolia	No plans	Mongolia aims to increase the incorporation of ICT in the classroom and has the general goal to increase the digital literacy of the entire population
North Korea	No plans	
South Korea	Implemented	Started with a nationwide pilot in 2015 and was implemented as a compulsory part of the curriculum for primary and secondary schools in 2018
Taiwan	Implemented	Began in August of 2019 to require every student in secondary school to be fostered with CT competences. This has so far been achieved by integrating CT into the curriculums of other courses

Taiwan has seen, like Denmark, an increased focus on CT and other digital literacy skills from the side of *clubs*. Moreover, many activities are currently conducted in Taiwan, to connect CT teachers with IT experts and learning researchers. We are cooperating with the "Mini Educational Game development group" (NTUST MEG[1]) at the National Taiwan University of Science and Technology in Taipei, to plan further collection of data and observations in-situ in the coming semesters. Thanks to our network, we are currently conducting interviews with teachers from schools and clubs

[1] Official web-site: http://www.ntustmeg.net/about.asp.

in Taiwan (mainly in Taipei), to better understand the complementary roles of formal and informal learning settings with respect to CT education. Semi-structured interviews were chosen (and carried out over VoIP) as this allowed salient points to be explored in depth as they occurred, helping us to cope with possible misunderstanding that can easily result from the language barrier or the use of VoIP. Starting with few open-ended questions, the interviews proceeded as a free-flowing dialogue between the interviewees and interviewer. The aim of the interviews is to gather insight from teachers about their motivation, and experiences with CT.

4 Discussion

According to our preliminary findings, despite the sociocultural differences in pedagogical practices in Taiwan and Denmark, the effort of defining and integrating CT as a curricular subject has raised shared questions regarding:

- The relation between formal and non-formal learning practices,
- The relation of CT with other subjects either science or art and design,
- The relevance and necessity of coding as a learning goal.

Given that the term CT originates from North American learning research (e.g. from projects like MIT's lifelong kindergarten, Scratch [9], original and current definition of CT like Wing's), it is perhaps not surprising that we found that many countries are looking at the USA and in particular MIT's tools like Scratch as a de-facto standard starting point for the discussion regarding CT. Therefore, we consider North American research to be an unofficial shared reference for the implementation of CT programs in East Asian and Scandinavian countries alike, as well as in Taiwan and in Denmark.

4.1 Formal and Non-formal Learning Practices in CT

According to our data, the implementation of CT in Scandinavia as well as in East Asia CT has led school teachers to experiment further with non-formal learning practices, integrating forms of playful learning and gamification. In the terms of Vygostky [22], this trend has translated into the orchestration of shared CT learning activities, in which CT have been associated with creative thinking, so that the pupils are asked to create simple games or interactive animations in tools like Scratch or Blockly, under the supervision of teachers or external experts acting as guest lecturers. In this way, the pupils are encouraged to engage with coding in a playful and exploratory way, so that through the design process the pupils engage in conceptual thinking regarding how the choices they make for their code will affect their resulting game [22]. This way of working enables pupils to engage with peer-learning as well as gaining supervision from teachers, in relation to self-motivated development processes, as in a form of apprenticeship [23].

In both Taiwan and Denmark, primary schools are experimenting with short mini-courses in CT, lasting for about a week or two and held by external teachers. These teachers might be researchers from academic institutions or experts from IT companies, who are testing new tools or pedagogical approaches to CT. During a typical mini course the pupils would be required to develop a game in Scratch or would be introduced to programming via Minecraft[2] within a series of 1-day workshops across a few weeks. Among IT experts and academics, PhD students from the University of Southern Denmark were sent to primary schools to hold CT workshops on Minecraft and Scratch. Informal discussion with these PhD students suggests that interacting with a class of pupils and with teachers of varying degree of IT competences may produce inconsistent results in relation to children's learning, as they might be subject to inconsistent and discontinuous forms of training.

Interestingly, coding alone is not the central focus of these workshops, but instead programming is introduced, often via design processes, involving sketching, development, and testing. Similar forms of playful learning are typically adopted in afternoon clubs in both countries, where children are being introduced to CT through workshop-like activities lasting for a semester or a year [6]. Hence in both Danish and Taiwanese clubs, pupils are supposed to develop games, combining coding with design thinking. In this sense, in spite of cultural differences, both countries are exploring a non-formal learning approach to the learning of CT.

Being CT a new subject in formation, it is not clear yet if it might be integrated in other existing subjects and which subjects these might be. CT has been seen in this regard as belonging to Mathematics and Natural Sciences, since the theoretical foundation of CT is rooted in Mathematics and Computer Science. However, CT could also belong with Arts and Design, since the application of CT deals with the design and realization of digital artefacts (often, but not necessarily, of games). On the other hand, it has been proposed that the analytical mindset from CT [10] can be applied to any subject, including for instance text analysis in humanities. This, however, can cause issues regarding the background of teachers formed in the humanities, as they might be less trained in understanding the core elements of CT and algorithmic problem solving than teachers formed within sciences or design related subjects. Moreover, the introduction of CT in existing subjects might lead to redefining the curricula of those subjects, with a risk of compromising the core content of the same subjects, hence, a consensus has not been reached in this regard in neither countries. However, according to our data, it seems that in both countries CT is being increasingly linked to Arts and Design than Mathematics and the Natural Sciences. In fact CT is presented as a collection skills and activities such as: coding, hardware, game design, and 3D printing, and leveraging on design thinking as an interdisciplinary framework providing the backbone of CT learning activities. Moreover, design processes oriented towards interactive media and games are seen as more effective motivational resources for children, than activities related to mathematics, a subject notoriously perceived as challenging.

According to our data, informal and playful learning activities appear as dominant in both schools and clubs, moreover, it is increasingly the case that children have experienced CT before they encounter it in schools participating in clubs informal learning activities. Hence, there might be a need of a dialogue across non-formal and

[2] Official web-site for Minecraft: www.minecraft.net/.

formal institutions in both countries. Moreover, the broad use of non-formal learning in CT has also implication for pedagogical alignment (see Biggs [8]) and assessment in schools, hence teachers are experimenting with different ways to test their students' progress, moving towards project-oriented assignments, in which pupils are asked to develop different artefacts, to gather them in portfolios documenting the development process, and to present to the class and the teachers. In Denmark project-oriented assignments are common also in schools and are applied to different curricular subjects. Interestingly, portfolio evaluations (such as those adopted for CT) are graded as written or oral examinations, typically in a qualitative way; however, subjects like Mathematics usually prefer quantitative evaluations. We find that these experiments with assessment are interesting to follow and are necessary, but they emphasize the complexity of placing CT in the curriculum. Central aspects of project-oriented assignments in both countries are group work and peer-learning, which are integrated in the pedagogical approaches of Danish schools as well as clubs. Similarly in East Asia we have found an increasing adoption of gamification approaches, which can be interpreted as an attempt to open up the classroom to forms of apprenticeship-like teaching, where pupils gain points by taking more responsibilities in their own learning (in line with Biggs and active learning [8, 23]). In this sense, the quest of turning CT into a curricular subject is similarly challenging current learning practices and cultures across East Asia and Scandinavia, encouraging an increasing adoption of informal learning approaches, bridging formal and non-formal learning contexts.

Finally, according to our literature review and our dialogue with academics in Taiwan and Japan, we found that both East Asia and Scandinavia are looking at the USA (and the MIT in particular) as leading forces regarding pedagogical approaches and e-learning tools. As a result countries in both regions are adopting books by leading American academics directly or through translations and adaptations, and a tool chain typically based on Scratch, Python and AppInventor. Also the adoption of gamification for technically difficult subjects, the notion of re-conceptualization and re-use of games (e.g. chess for Math, microwords [24] for exploring various domains, etc.) for learning purposes, are inspired by North American research, although reframed within our respective pedagogical traditions. In this respect, in spite of cultural differences, we find that the implementation of CT as a school subject in Denmark and Taiwan is converging towards similar pedagogical models and the adoption of the same digital tools, inspired by the North American context. The emergence of CT as a school subject is challenging local learning cultures, forcing a more global perspective in education than other subjects with a longer didactic tradition like, for instance national languages or History.

4.2 The Role of Coding in CT

Another common challenge we identify from both East-Asian and Scandinavian perspectives regards the importance of programming as a learning goal, and even when focusing only on coding, as a part of programming, it is not easy to re-scale coding to fit the needs of primary school pupils.

Although we tend to agree that CT is not just programming and that coding is a subset of programming, all the approaches we have encountered consider programming (or at least coding) as a desirable but challenging learning goal of CT. In fact, most e-learning tools for CT are de facto aiming in different ways at introducing primary or secondary school pupils to programming [25]. This trend is due to the perception of programming as a precious 21st century skill in the global economy, providing an advantage in the job market to technical as well as humanistic graduates [11]. However, programming itself is not yet a school subject in many countries, not at the high school level, nor in primary schools. Therefore, CT appears as a holistic strategy to enable pupils to get closer to programming through creativity and applied problem-solving, avoiding the challenge of introducing in primary and secondary education Computer Science-related mathematical formalisms, which are found hard by technical students at the university level.

As many studies are simultaneously being conducted in different countries on CT, specifically on the definition of pedagogical approaches and curricula [10] and also on the development of e-learning tools, such as: games, block coding editors, or simplified hardware kits [6]. Hence an increasing number of primary and secondary schools are experimenting with teaching CT through courses or short workshops, led by teachers or external guests. As a result, we find that schools as institutions and individual teachers are experiencing increasing pressure in framing programming within their curricula and in finding proper tools, to enable their pupils to engage with and acquire CT knowledge and skills. In this respect, we find programming as the central challenge across Scandinavia and East Asia, requiring pedagogical framing as well as support from adequate e-learning tools, addressing the needs of pupils and teachers.

We have, therefore, surveyed existing tools for coding in primary schools and found (in line with [16]) that existing IDE for beginners, including the popular Scratch, MU editor, and Blockly-based visual IDEs, are conceptually created with a single user in mind: in general, coding tools assume a lone programmer. In the terms of Vygotsky [22] and Rogoff [23], we see that coding tools embody a perception of coding as an individual activity, therefore, they might not support adequately current orchestration strategies adopted in CT learning activities, which leverage forms of peer-learning and collaborative problem-solving. The computer itself is thought to afford individual use at the keyboard. Perhaps to compensate this lack of support, we find that CT activities are implicitly taking the form of an apprenticeship [23] supported by design thinking and peer-learning in which pupils are given group activities, to be further segmented into individual coding tasks. Game design is often chosen as non-formal, playful activity to encourage pupils to engage with programming and coding, shifting between individual and collaborative tasks.

Typical unsolved problems for the teachers in this area are addressing learners with substantially different skills in the same class, as it can be the case of a class mixing beginners and pupils with prior experience in CT from clubs, or when introductory courses mix students from different educations (as discussed in [12]). In such cases, teachers must be able to provide an activity suitable to pupils with different proficiency levels and avoid the programming version of the "alpha player problem" known in cooperative games, i.e. the tendency of one participant, not necessarily the most expert, to take over an activity and makes cooperation impossible for others. As a result we find that the learners who have the most need to practice will not practice enough to overcome their difficulties, while the stronger programmers in the group are those who practice the most, becoming even more proficient (a problem discussed also in the context of introductory programming courses at university level [12]). The need for planning and structure programming as a social practice is embodied in tools like Trello (trello.com), aimed at planning the activities of Scrum-teams, and platforms like Github provide not only ways to share the codebase, but also networking and management support for developers. However, these tools are typically aimed at advanced and older programmers, and have not been developed with primary school teachers and pupils in mind. Moreover, in professional "peer programming practices", there are well known roles that programmers can take at different stages of development, which are not explicitly supported by coding tools for pupils. Building on these insights, we identify a urgent need for coding tools, accessible to pupils, that regard coding as a social practice and learning to code as an apprenticeship, supporting small teams of learners in their process of crafting code and not just in dividing tasks.

5 Which E-learning Tools to Support CT?

Building on Vygotsky learning theory [22] as well as our preliminary data, our main contribution is the definition of a global-informed scenario (based on [2, 7] and [6]) and requirements for a new class of e-learning tools, capable of supporting emergent strategies in the learning of CT from a global perspective, such as:

- Support for peer-learning in small groups of pupils, leverage on Worked Examples in the form of functioning programs,
- Rely on the Use-Modify-Create approach, for instance by allowing pupils to work together at altering and adding to existing code,
- Support communication among learners and with teachers,
- Involve multiple subjects in the provided programs to allow discussions about technology in society.

As a domain for the provided functioning programs, we consider simple digital games, easy to reprogram, possibly expressed in simplified ways, such as blocks. In fact, building on our data, we find that game design has been adopted as a main domain for the learning of CT in East Asia as well as in Scandinavia, taking inspiration from the North American context, where popular tools like Scratch or Minecraft are specifically designed for supporting children in developing and altering games. Games are seen as fostering pupils' self-motivation, leveraging their interest for games in their everyday life, but also as a concrete domain enabling pupils to envision their final product, to make plans for their code, and the necessary steps to develop such product. In this respect, game development provides an apprenticeship framework, characterized by the goal of creating a game, in parallel to the goal of learning CT, in which kids and adults are mutually engaged [23]. Moreover, in game development as in apprenticeship pupils can learn from each other within their groups and also across other groups, developing similar games, and are allowed to gradually take over responsibility through their process, by planning their design process and distributing tasks with each other. In this way, learning to programming becomes an apprenticeship activity, in which learners are allowed to engage independently with the available coding tools and in making decisions on their process, while at the same time receiving support when reaching concepts and practices on the boundary of their Zone of Proximal Development [22, 23].

The scenario in Fig. 1 is based on [7] and inspired by the "Fabric Robotics" approach described in [6], and follows the Use-Modify-Create paradigm. In Fig. 1 the teacher has divided 5 pupils in 2 groups, and each group starts by using an existing project (possibly a simple game implemented in a block-based language). The group on the left in Fig. 1 is modifying their game and submits a new version to the teacher (or documentation that explains their design and implementation process), the other group (composed of 3 "green" pupils) is instead creating a completely new game, after having played the assigned one.

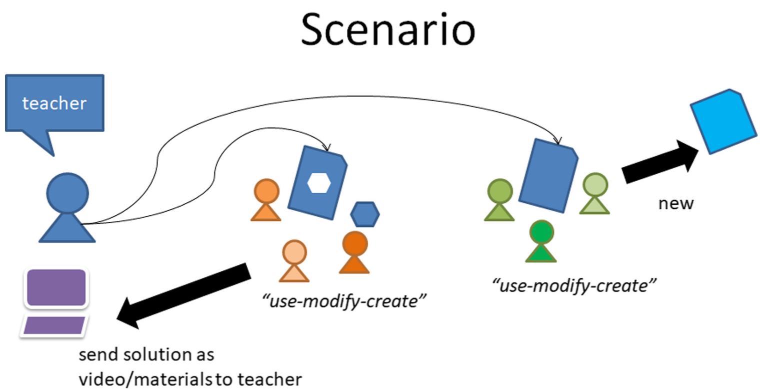

Fig. 1. Visualization of our scenario, with a teacher and learners working in small groups. (Color figure online)

According to these requirements and related learning scenario, the resulting tool would be a system to create and share digital games, easy to reprogram, and providing some form of real-time as well as asynchronous communication capabilities for collaborative learning. This tool should support the scenario in Fig. 1 and could for example be implemented as a generator of Jupyter Notebooks, specific to simple digital games and with block-based code instead of the usual textual commands. In our view, the ideal tool for supporting CT learning should embody the values and relevant functionalities to support an Apprenticeship learning scenario.

Moreover, we can see that in practice both Denmark and Taiwan are adopting a Use-Modify-Create approach, even if only Denmark is explicitly referring to it by name, converging towards constructivism and playful learning. The adoption of this approach might be due to the predominance of constructivist discourse in North American research on CT, which is implicitly informing other countries when they adopt the materials from MIT and similar institutions. In the Use-Modify-Create paradigm, children spontaneously engage in forms of conceptual thinking by playing with an artefact [22], like a computer game in CT activities, and that can enable them to reflect on the hypothetical changes they might perform on the given game and on its code. This approach can be seen as a pedagogical concretization of the concept of proximal development [22], according to which pupils need step by step support, from using to edit software, and finally to become creators of code, the final step in overcoming the challenges faced while reaching the boundary of their CT abilities. This progression also embodies the values from Apprenticeship learning, in which the pupils are supposed to become more in charge of their process, starting with full supervision, in this case translated into use of a given game, to end with independence and the ability to create a new game [22]. Finally this progression can be interpreted as an adaptation of Bloom taxonomy [26] to games-based learning practice for CT, in which the pupils are supposed to gradually reach the ability to use their knowledge to create something new. Based on these insights, we propose that coding tools for learning CT should embody this progression in their design and in the examples provided to the pupils, so to enable them to gradually engage with more complex tasks, in respect on their gradual process of learning to master coding (and possibly also programming).

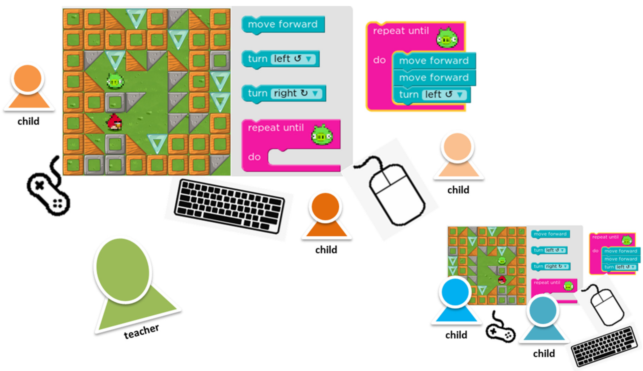

Fig. 2. A possible scenario of use for our new tool. Two groups of pupils work at the same game, with a teacher supervising both groups.

Figure 2 shows a possible situation where 2 groups of pupils are working on the same game, that they received from the CT teacher. Our new tool is used to distribute the game code and for communication among the groups and with the teacher, as well as for delivery of modified or new versions of the game. We propose that each child in Fig. 2 should have a different, explicit role while using the new tool. In the group with 3 “orange” pupils for example the left-most child is play-testing the game using a game controller, while at the same time talking with the child in the middle, who is close to a keyboard and therefore acts as the developer, altering the code to accommodate the feedback of the play-tested child. The third member of the group, on the right, could be using the mouse to take snapshots of interesting moments to document the gameplay as it changes because of the interaction of the tester and the coder.

6 Conclusion

This paper analyzes the emerging pedagogy of Computational Thinking as a new school subject, from an international perspective. Approaches from Scandinavian and East Asian are compared, focusing on Denmark and Taiwan. Preliminary findings from our analysis and initial interviews show that, despite sociocultural differences in pedagogical cultures, both countries are facing similar challenges related to the relationship between formal and non-formal learning practices, the positioning of CT with respect to other school subjects, and the effort to adapt North American discourse to the local educational traditions.

Coding emerges as a highly desirable and necessary activity within CT, as well as a central learning goal. One of the contributions of this paper is the identification of a lack of coding tools for small groups of pupils as a gap in CT e-learning. The other major contribution is the definition of a scenario and requirements for a new class of e-learning coding tools for better supporting emergent strategies in the learning of CT, such as: peer-learning, group-work especially for small groups of pupils, take advantage of pedagogical techniques like Use-Modify-Create and Worked Examples.

Future work involves cooperation with our contacts in Denmark, Taiwan and Japan to observe their practices in teaching programming within CT. We are currently developing, together with our international network, a prototype of a new e-learning tool, that will be tested in primary schools in both Scandinavia and East Asia.

References

1. Huang, C., Lin, H., Wang, S., Hou, H.: Designing a gamified activity with visual representation-based scenario and technology-based scaffoldings for learning electric potential. In: IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), Yilan, pp. 1–2 (2019). https://doi.org/10.1109/ICCE-TW46550.2019.8991794.
2. Marchetti, E., Valente, A.: It takes three: re-contextualizing game-based learning among teachers, developers and learners. In: Connolly, T., Boyle, L. (eds.) Proceedings of The 10th European Conference on Games Based Learning, pp. 399–406. Academic Conferences and Publishing International (2016)
3. Kristensen, K.: Towards computational thinking in Scandinavia. In: 27th International Conference on Computers in Education, ICCE 2019, pp. 47–49. Asia-Pacific Society for Computers in Education (2019)
4. Kristensen, K.: Implementation of computational thinking in school curriculums across Asia. In: Stephanidis, C., Antona, M. (eds.) HCII 2020. CCIS, vol. 1225, pp. 269–276. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50729-9_38
5. Bocconi, S., Chioccariello, A., Earp, J.: The Nordic Approach to introducing Computational Thinking and programming in compulsory education. Report promoted and funded by the Nordic@BETT2018 Steering Group (2018). https://doi.org/10.17471/54007
6. Pedersen, B.K.M.K., Marchetti, E., Valente, A., Nielsen, J.: Fabric robotics - Lessons learned introducing soft robotics in a computational thinking course for children. In: Zaphiris, P., Ioannou, A. (eds.) HCII 2020. LNCS, vol. 12206, pp. 499–519. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50506-6_34

7. Valente, A., Marchetti, E.: From cards to digital games: closing the loop. In: 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 507–510. IEEE (2017)
8. Biggs, J.B.: Teaching for Quality Learning at University: What the Student Does. McGraw-Hill Education, New York (2011)
9. Resnick, M.: The next generation of Scratch teaches more than coding (2019). https://www.edsurge.com/news/2019-01-03-mitch-resnick-the-next-generation-of-scratch-teaches-more-than-coding
10. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
11. Marchetti, E.: Bildung and the digital revolution. In: The University of the Future. Academic Conferences and Publishing International (2019)
12. Valente, A., Marchetti, E., Wang, J.: Design of an educational multimedia library to teach Python to non-technical university students. In: LCT: 6th International Conference on Learning and Collaboration Technologies. Springer, New York (2020). https://doi.org/10.1109/IIAI-AAI50415.2020.00041
13. Gee, J.P.: What Video Games Have to Teach us About Learning and Literacy. Palgrave Macmillan, New York (2007)
14. McGonigal, J.: Reality is broken: Why games make us better and how they can change the world. Penguin (2011)
15. Hsu, T.-C.: A study of the readiness of implementing computational thinking in compulsory education in Taiwan. In: Kong, S.-C., Abelson, H. (eds.) Computational Thinking Education, pp. 295–314. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-6528-7_17
16. Hsu, T.C., Chang, S.C., Hung, Y.T.: How to learn and how to teach computational thinking Suggestions based on a review of the literature. Comput. Educ. **126**, 296–310 (2018)
17. UVM. Informatik C, hhx, htx, stx, hf Vejledning (2019). https://www.uvm.dk/gymnasiale-uddannelser/fag-og-laereplaner/laereplaner-2017/hhx-laereplaner-2017. Accessed 25 Feb 2021
18. UVM. Undervisningsvejledning for forsøgsfaget teknologiforståelse (2020). https://www.uvm.dk/-/media/filer/uvm/publikationer/folkeskolen/faghaefter/faelles-maal-2014/200813-teknologiforstaaelse-undervisningsvejledning.pdf. Accessed 25 Feb 2021
19. Atkinson, R.K., Derry, S.J., Renkl, A., Wortham, D.: Learning from examples: instructional principles from the worked examples research. Rev. Educ. Res. **70**(2), 181–214 (2000)
20. Caspersen, M.E., Nowack, P.: Computational thinking and practice a generic approach to computing in danish high schools. In: Proceedings of the 15th Australasian Computing Education Conference, ACE 2013, Adelaide, pp. 137–143 (2013)
21. Lee, I., et al.: Computational thinking for youth in practice. ACM Inroads **2**(1), 32–37 (2011)
22. Vygotsky, L.S.: Mind in Society. The Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)
23. Rogoff, B.: Apprenticeship in Thinking: Cognitive Development in Social Context. Oxford University Press, Oxford (1990)
24. Resnick, M., Silverman, B.: Some reflections on designing construction kits for kids. In: Proceedings of the 2005 Conference on Interaction Design and Children, pp. 117–122 (2005)
25. Sorva, J., Karavirta, V., Malmi, L.: A review of generic program visualization systems for introductory programming education. ACM Trans. Comput. Educ. (TOCE). **13**(4), 1–64 (2013)
26. Selby, C.C.: “Relationships: computational thinking, pedagogy of programming, and Bloom’s Taxonomy. In: Proceedings of the Workshop in Primary and Secondary Computing Education, pp. 80–87 (2015)

27. Børneog: Undervisningsministeriet. Midtvejsevaluering: Forsøg med teknologiforståelse fremmer faglighed og kompetencer (2020). https://xn–tekforsget-6cb.dk/wp-content/uploads/2020/05/Midtvejsevaluering-Maj-2020.pdf. Accessed 25 Feb 2021
28. Johnson, D.W., Johnson, R.T.: Cooperative learning. In: The Encyclopedia of Peace Psychology (2011)
29. van de Oudeweetering, K., Voogt, J.: Teachers' conceptualization and enactment of twenty-first century competences: exploring dimensions for new curricula. The Curriculum J. **29**(1), 116–133 (2018)
30. Tedre, M., Denning, P.J.: The long quest for computational thinking. In: Proceedings of the 16th Koli Calling Conference on Computing Education Research, Koli, pp. 120–129 (2016)
31. Lave, J., Wenger, E.: Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, Cambridge (1991)
32. Smith, P.K., Cowie, H., Blades, M.: Understanding Children's Development. Blackwell Publishing, Hoboken (2003)