A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K–12 Education

Fredrik Heintz
Dept. of Computer Science
Linköping University
Email: fredrik.heintz@liu.se

Linda Mannila
Dept. of Computer Science
Åbo Academy and Linköping University
Email: linda.mannila@abo.fi

Tommy Färnqvist
Dept. of Computer Science
Linköping University
Email: tommy.farnqvist@liu.se

***Abstract*—Computing is becoming ever increasingly important to our society. However, computing in primary and secondary education has not been well developed. Computing has traditionally been primarily a university level discipline and there are no widely accepted general standards for what computing at K–12 level entails. Also, as the interest in this area is rather new, the amount of research conducted in the field is still limited. In this paper we review how 10 different countries have approached introducing computing into their K–12 education. The countries are Australia, England, Estonia, Finland, New Zealand, Norway, Sweden, South Korea, Poland and USA.**

The studied countries either emphasize digital competencies together with programming or the broader subject of computing or computer science. Computational thinking is rarely mentioned explicitly, but the ideas are often included in some form. The most common model is to make it compulsory in primary school and elective in secondary school. A few countries have made it compulsory in both. While some countries have only introduced it in secondary school.

I. Introduction

In recent years, we have witnessed an active discussion surrounding the role of programming and computing for everyone (see e.g., [1], [2]. As a result, an increasing number of countries have introduced or are in the process of introducing computing in their school curriculum. For instance in Europe, the majority of countries (17 out of 21) taking part in a survey conducted by the European Schoolnet reported doing so [3]. The way in which this is accomplished varies. Some countries focus on K–12 as a whole, whereas others primarily address either K–9 or grades 10–12. Some countries have introduced computing as a subject of its own (e.g. England [4]) while others have decided to integrate it with other subjects, by for instance making programming an interdisciplinary element to appear throughout the curriculum.

While the former approach seems like the most straightforward alternative, there are different reasons for why some countries are opting for the latter alternative: 1) lack of space for introducing a new subject in the curriculum, 2) letting students see and experience the use of programming in different subjects is seen as a way of raising interest among previously underrepresented groups, and 3) computational thinking is considered more and more important in all areas of society, and provides us with a framework for how we can work together with the computer to solve increasingly complex problems. As computing has traditionally been primarily a university level discipline, there are no widely accepted general standards for what computing at K–12 level entails. Also, as the interest in this area is rather new, the amount of research conducted in the field is still limited.

For this article we have conducted a review of different models for introducing computing (programming, computational thinking) in K–12 education. The aim is to synthesize current practice and thereby contribute to the general understanding for different approaches to introduce computing at school. We have focused on curricula or plans in the following countries: Australia, England, Estonia, Finland, New Zealand, Norway, Sweden, South Korea, Poland and USA.

For each country, we briefly describe the local educational system, the current status of computing in the curriculum and the current status of teacher training. We are interested in understanding different models for introducing computing or some aspect of computing in K–12 education. Important questions are: What is being introduced? In what subjects? What does the progression look like throughout the education system? What are the desired effects of the changes? Which teachers are responsible for the new content? What type of professional development is provided to the teachers? Are teacher education programmes affected by the changes? How are the efforts expected to be evaluated? How is teaching material being developed?

The studied countries either emphasize digital competencies together with programming or the broader subject of computing or computer science. Computational thinking is rarely mentioned explicitly, but the ideas are often included in some form. The most common model is to make it compulsory in primary school and elective in secondary school. A few countries have made it compulsory in both. While some countries have only introduced it in secondary school.

A. Related Work

There has been a few overviews related to K–12 computer science education. Grover and Pea reviews research on computational thinking in K–12 [5]. Mannila et. al. gives an overview of activities in K–9 [6]. Baarendsen et. al. gives an overview of concepts used in K–9 education [7]. Hubwieser et. al. [8], [9] gives a bird's eye view of K–12 computer science education.

Besides the actual models for introducing computing in K–12 education, it is also an interesting question how the academic community in the respective country studies these models and publish scientific articles concerning them. To shed a little light on this issue, we looked at the proceedings of a few selected conferences: Australasian Computing Education Conference (ACE), 2005–2016, International Workshop on Computing Education Research (ICER), 2005–2015, International Conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP), 2005, 2006, 2008, 2010, 2011, 2013–2015, Integrating Technology into Computer Science Education (ITiCSE), 2005–2015, and Workshop in Primary and Secondary Computing Education (WiPSCE), 2012–2015.

Table I shows, for each country and year, whether a conference paper has been published that concerns design of K–12 computing curricula, whether computing should be integrated or a subject of its own, or teacher training.

II. Terminology

Whereas Computer Science is the established academic term for the scientific discipline underlying the current digitalization and information technology, there is no clear agreement on what term to use at lower levels of education.

Computer science, computing and informatics refer to more or less the same thing, that is, the entire discipline.

Programming and coding are commonly used as synonyms. The European Schoolnet report "Computing our future" [3], showed that most European countries use programming or coding in their curriculum.

Furthermore, computational thinking, a term coined by Papert in 1996 [81], and gaining traction in 2006, through a seminal article by Wing [82], has seen a large increase in popularity during recent years. There is no agreed upon definition of computational thinking, but it builds on "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be carried out by an information-processing agent" [83].

All of the terms mentioned above differ greatly from information (and communication) technology (IT, ICT), which mostly focus on computer literacy, that is, knowing how to use computers and their applications as tools.

In the following, we will use *computer science* as the umbrella term when discussing the situation in general. When referring to specific countries, we will use the term used in their documents.

III. Methodology

We decided to include ten countries in our study, covering Europe, USA, Asia and Australasia: Australia, England, Estonia, Finland, New Zealand, Norway, Sweden, South Korea, Poland and the USA. Some of these have introduced computer in the official curriculum, whereas others are in the process of doing so.

In order to study ways and models for how computer science can be introduced, we selected the following criteria to focus on:

- Term used (cf. the discussion in Section II above)
- Role in primary level curriculum (compulsory/elective/other)
- Role in secondary level curriculum (compulsory/elective/other)
- Integrated or a subject of its own
- Suggested progression
- Teachers responsible for covering the material
- Teacher training (in-service and pre-service)
- Teacher material

The study was done by reviewing relevant documents for each country. For those, who have already introduced computing in their curriculum, we focused on the corresponding official documents. For countries, where no official guidelines have yet come into force, we used other documents in the form of, for instance, articles and white papers.

The review was done according to the principles of content analysis [84]. The basic idea of content analysis is to take textual material and analyze, reduce and summarize it according to pre-defined or emergent themes. We used the criteria above as the basis for analysis and reported on our findings in a table. Each author reviewed 3–4 countries. In situations where somebody was indecisive about how to report on a given matter, all authors jointly discussed the topic at hand.

An alternative model is the Darmstadt Model[85]. However, this is a framework for research in computer science education, rather than a framework for comparing national curricula and ways of introducing computer science into primary and secondary education.

A. Australia

The Australian primary and secondary school system is undergoing a significant period of change, with the introduction of a National Curriculum [11]. In 2015, Australia [86] endorsed the Australian Curriculum: Technologies [87] that incorporates both Digital Technologies and Design and Technology. Within Digital Technologies (DT), children are to develop computational thinking skills and learn about data, digital systems and how to implement solutions with programming. We note that DT in the Australian curriculum is a "learning area" in its own right, on par with English, Mathematics, Science, Humanities and Social Sciences, The Arts, Health and Physical Education, and Languages. It is also apparent that ICT capabilities, as a whole, correspond to learning objectives in the DT learning area as well as objectives spread out over all other learning areas.

In Australia, primary school includes the first year of school, called Foundation (F) followed by year 1, and so on, until year 6 or 7 (depending on the state) and secondary school (also known as high school) includes years 7 or 8 to year 12. The curriculum learning objectives are organised around a series of

TABLE I
PUBLISHED PAPERS AT SELECTED CONFERENCES FOR EACH COUNTRY AND YEAR.

	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Australia		[10]								[11]	[12]
Finland		[13]									
Great Britain	[14]	[15]				[16]		[17]	[18]		[19], [20], [21]
New Zealand						[22]		[23]	[24], [25]		[26]
South Korea		[28]									
Sweden											[29]
Poland	[30], [31]			[32]					[33]		[34]
USA	[35]		[36]		[37]	[38], [39], [40]	[41]	[42], [43]		[44], [45]	[46], [47]
Austria	[48], [49], [50], [51], [52]	[53]					[54]	[55]			
Czech Republic								[56]			
Denmark									[57]		
France								[58]	[59]		
Germany		[60]			[61]	[62]		[63]	[64]		
Greece								[65]			
Hungary		[66]									
Israel		[67]					[68]	[69]	[70]		
Italy		[71]					[72]			[73]	
Lithuania	[74]	[75]		[76]							
The Netherlands				[77]					[78]		
Russia	[79]										
Switzerland							[80]				

Year level “bands”, from F to Year 10, with senior year levels awaiting final development. While junior year objectives are mandatory, senior year students may select specialized strands in the DT learning area.

The DT curriculum has a strong focus on computational thinking skills and the development of both digital literacy and computational thinking commences in the F–2 band and learning is based around directed play, facilitating students in developing an understanding of the relationship between real and virtual worlds, the use and purpose of technology in communication and the importance of precise instructions and simple problem-solving in the digital world. In years 3–6, students are guided to develop a wider understanding of the impact of technology, including family and community considerations, and are able to work on, and communicate about, more complex and elaborate problems. In this year level, students begin to develop algorithms with visual programming software. Across years 7–10, students move beyond their initial community and are required to consider broader ethical and societal considerations. Students solve more sophisticated problems using technology, and develop an understanding of complex and abstract processes. Students use programming languages to solve problems and create digital solutions.

Australian primary school teachers are typically generalist teachers, trained to teach across the various learning areas prescribed by their state or territory, with only 6% (n=7,500) currently teaching computer science [88]. To provide teachers with appropriate support, initiatives such as a professional development MOOC [88] and a systematic review of CS resources fit for the DT curricula [89] have been instigated.

B. *England*

In England, a new national curriculum came into force in 2014 [4], [90], introducing a new subject, Computing, replacing the previous ICT curriculum. England is thus one of the few countries, which is not focusing on programming as an integrated trait, but rather on the larger discipline as its own subject.

The subject Computing contains three elements: computer science, information technology and digital literacy. The main aims are to ensure that all students:[4]

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology

The way in which these aims are to be reached differs at the four key stages, covering both primary and secondary education.

Computing at school (CAS, http://computingatschool.org.uk) is a community promoting computing science at school. The community has (April 2016) over 22k registered users, nearly 75k discussion posts, over 190 local hubs and nearly 3500 teaching resources.

CAS hubs are meetings of teachers and lecturers who wish to share ideas about teaching computing in their schools, classrooms, and communities. The hub network has been credited in part with the success of the CAS project in that teacher isolation is reduced and teachers become more energetic after participating in activities and meeting organized by the hubs.

The teaching resources include lesson plans and guidelines for different levels, starting with Barefoot Computing (http://barefootcas.org.uk) and Teach Primary Computing (http://teachprimarycomputing.org.uk) for primary educators and QuickStart Computing (http://quickstartcomputing.org) providing support for both primary and secondary level. In addition, passionate teachers have also developed and shared teaching material (see, for instance, http://code-it.co.uk).

CAS also offers accreditation for Computing teachers, providing professional recognition by the British Computing Society. The certificate includes three parts: 1) reflection on professional development, 2) programming project and 3) classroom investigation.

In addition, there is a Network of Teaching Excellence in Computer Science gathering professionals wanting to work together on these matters and providing widespread professional development. For example, the network includes CAS Master Teachers, who are experienced teachers with a particular passion for the subject and for supporting others.

C. Estonia

The Tiger Leap Foundation (TLF) is an Estonian organization, founded in 1997, aiming at developing the country's IT infrastructure and the quality of education with the help of technology [91]. As part of the results all Estonian schools were connected to the Internet, teachers were trained in computer usage and schools were provided with computers.

The ProgeTiger programme [92] was launched in 2012, aiming at introducing programming and robotics in education. The programme is carried out by the HITSA Development Centre of IT Education, funded by the Estonian Ministry of Education and Research. Other partners include the Estonian Ministry of Economic Affairs and Communications, as well as universities, the private sector and third sector institutions. aimed at pre-school, primary and vocational education and talks about digital competence as including both skills in using technology efficiently and responsibly, as well as understanding the essence of technology and being able to create technology.

According to HITSA [92], this competence is described as follows in the national curriculum for primary education: "the ability to cope in the technological world, understand technology trends and the connections between technology and other scientific achievements; to acquire technological literacy for age-appropriate, creative and innovative use of technology tools, integrating thinking with manual activities; to analyse opportunities and risks associated with the implementation of technology; to comply with the requirements for intellectual property protection; to solve problems by integrating thinking with manual activities and carry out ideas purposefully; to cope with household chores and eat healthy."

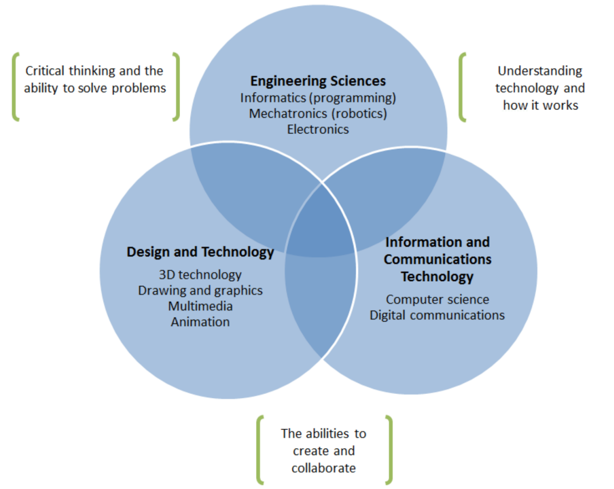

Fig. 1. The ProgeTiger model in Estonia. Source: [92]

The ProgeTiger programme integrates three thematic fields (see Fig. 1) — engineering sciences, design and technology and ICT — with different subjects. Engineering sciences include programming, electronics and robotics, while Design and Technology covers, for instance, 3D modeling, graphics, multimedia and animations focusing on user-friendly design. ICT focuses on the use of computers, networks and technology tools. To meet the goals of the programme, HITSA develops learning materials and offers professional development courses for teachers.

D. Finland

A new national curricula will come into force in Finland in 2016, covering both primary (grades 1–9) [93] and secondary (grades 10–12) [94] education. The curricula put increased focus on digital competence as an interdisciplinary trait throughout all grades. As for computer science content, the Finnish curriculum includes programming as an integrated element in primary education, while no computer science content is mentioned in the curriculum for secondary level.

At primary level, programming is mentioned explicitly in mathematics for grades 1–2, and in mathematics and craft for grades 3–9. In addition, programming is included as part of digital competence covering all subjects — hence, programming can be introduced in all subjects. In grades 1–6, teachers teach all subjects, which consequently means that all teachers are affected by the change. In grades 7–9, teachers teach 2–3 subjects, which may result in mainly those teaching mathematics and craft teaching programming.

In Finland, the National Board of Education, as well as the Ministry of Education and Culture fund projects, professional development and development programmes for supporting teachers in providing ways and materials for integrating programming in teaching and learning. In addition to state supported initiatives, various activities are also provided by private actors, universities and organizations.

In 2014, the Ministry of Education and Culture published a call for specialization training programmes for in-service teachers. One of the accepted programmes was on teaching and learning in digital environments. This programme is offered, for instance by the University of Turku in Finland, including general courses on the topic, but also specialized courses on programming and making.

E. New Zeeland

Computer science has been a subject of its own the last three years of high school since 2011. Previously the focus was on teaching how to use computers [95]. "Programming and computer science" is a strand of Digital Technologies (http://www.nzqa.govt.nz/ncea/assessment/search.do?query=Digital+Technologies). It contains both programming and a broad range of computer science topics, including algorithms, human-computer interaction, artificial intelligence and computer graphics,. These topics are not covered in depth, but rather cursory. This gives the students an understanding what computer science is about. The topics are loosely based on the ACM/IEEE computer science curriculum [96].

The progression in programming starts with introductory work in year 10, through the equivalent of an introductory university course in year 12 [95]. Year 10 focuses on tasks that involve input and output, and can be expressed as a single procedure program using sequence, selection and iteration, but only requires simple data (no arrays, lists, or structures). Year 11 focuses on tasks that involve multiple procedures and also use an indexed data structure. Year 12 requires the use of basic object-oriented programming concepts (classes and objects with encapsulation, but not inheritance) and a simple GUI implementation with event handling. For year 10 and 11, graphical programming languages like Scratch are allowed. For year 12 a text-based programming language is required. Many school introduce text-based programming already in year 11, with Python as the most popular choice.

Programming and computer science is also formally part of the National Certificate in Educational Achievement (NCEA), the main school-leaving assessment [95].

The new curriculum was introduced very quickly with significant challenges to prepare the teachers. As support a post-graduate distance course was developed to allow teachers to obtain a formal qualification in teaching computer science. The work on CS Unplugged [97] that started in the 1990s had developed experience in communicating computer science without using computers, and in environments where little time was available to learn programming [95].

To provide suitable teaching material, which was identified as very important by the teachers, a free online open-source "CS Field Guide" was developed (http://csfieldguide.org.nz/en/about.html). It is an interactive site that is being developed to provide information at the level required for the new computer science standards, including notes for teachers.

Educating school management is an ongoing challenge, and is an important element for the success of computer science in schools.

F. Norway

Norway has not yet introducing computing into their curriculum, instead they will start a large pilot study in 146 secondary schools in the fall of 2016 (https://www.regjeringen.no/no/aktuelt/koding-blir-valgfag-pa-146-skoler/id2481962/). The schools will introduce programming as an elective for the students in year 8–10. The curriculum (http://www.udir.no/globalassets/filer/lareplan/forsok/forsokslareplan-programmering-som-valgfag.pdf) states that the purpose is to teach programming including identifying problems, developing solutions, systematic debugging and improving the code, and documenting the solution in an understandable way. The students should learn at least two programming languages, and at least one should be text-based. An important motivation is to increase the interest in technology.

To train the teachers a Massive Open Online Course (MOOC) has been prepared. To provide teaching material and also professional development an association called "Lær Kidsa Koding" has been started which has a site with extensive material (http://kidsakoder.no/).

G. South Korea

The Korean school system consists of 6 years of elementary school, 3 years of middle school, and 3 years of high school [98]. So called "computer education" started already in 1971. Since 2000, when the South Korean ICT infrastructure provided a computer to almost every classroom, contents related to computers became nearly mandatory, with more than 34 hours of instruction in each grade. In middle and high school curricula the subject was designated as elective. In 2007, "computer" was changed into "informatics" in the national curriculum, and the main focus has been on computer science principles and concepts.

A number of factors now seem to have worked together to make enrollment in informatics education in South Korea at middle and high school levels drop dramatically, from 80% to 23% and from 47% to 8%, respectively, between 2004 and 2012. Choi et al. [98] cite decreased time for elective subjects and the fact that various guidlines and policies concerning ICT education were repealed in 2008 as the main culprits.

In 2013, however, consensus concerning the importance of computer science education was reached, and, in 2018, a new curriculum will be introduced in South Korea [98]. It will consist of a compulsory subject "Informatics" in Middle School and an elective in High School. The curriculum will cover digital literacy, computational thinking and programming. Since the education system in South Korea is heavily textbook centered, new textbooks for the new curriculum is currently being developed.

H. Sweden

In 2015 the Swedish government gave the National Agency for Education (Skolverket) the task to develop a new curriculum for primary education (K–9) and to update the curriculum for secondary education (10–12). They explicitly stated that the curriculum should strengthen the digital competence of the pupils and it should introduce programming.

The National Agencey for Education has now presented its almost final proposal (http://www.skolverket.se/skolutveckling/resurser-for-larande/itiskolan/lamna-synpunker-1.246272). The final proposal is due June 30th 2016. If the government will accept the proposal and when it will start is still unclear. It is expected to start quite soon, maybe as soon as the fall 2017.

The proposal introduces a new general section on the importance of digital competence and understanding of digital technology for all and suggests changes to the course plans for math, technology and social sciences. Programming is introduced as part of the math curriculum. The progression is from giving step-by-step instructions as the basis for programming in grade 1–3, to how algorithms using sequence, alternative, repetition and abstraction can be created and used as part of programming in grade 4–6, to how algorithms can be created, tested and improved as part of programming for mathematical problem solving. Programming is also part of technology, where the focus is on controlling things. The understanding of computers and networks is also introduced in technology. The changes to social sciences focus on digital competence and becoming a critical and responsible digital citizen. The are also smaller changes to the natural sciences to introduce modeling and simulation into these subjects and also extending the range of materials to be used in crafts. The mandatory use of digital technology and tools is also introduced in several subjects.

Programming is already an elective in secondary school, however only available to a small selection of pupils in the natural science and technology programs. This not changed in the new proposal, but there is a new initiative to increase the access to programming and computer science in secondary education.

The National Agency for Education has also reveiced funding for national school development programs which will include in-service training for teachers. Currently teacher education does not include any mandatory courses in digital competence or programming. Since the National Agencey for Education has nothing to do with university education it is not clear how they will be influenced by the proposal.

Heintz et. al. [29] present an overview of ongoing activities related to computing and computational thinking around Sweden. This shows that is already today possible to introduce computing within the current curriculum.

A history of computing in the Swedish education was published in 2014 [99].

I. Poland

Computer science has been a subject in the Polish secondary education since the second half of the 1960's [100]. The first national curriculum was introduced in the second half of the 1990's.

In primary schools (grade 1–6) there is a stand-alone subject called computer activities. In grades 1–3, computer activities are supposed to be fully integrated with other activities like reading, writing, calculating, drawing, playing etc. At the next stages of education students are expected to use computers as tools supporting learning of various subjects and disciplines. The focus is on ICT.

In middle school (grade 7–9), informatics is taught for at least 2 hours per week for one year or one hour per week for two years. The curriculum contains algorithms, algorithmic thinking and problem solving with computers. Programming is not explicitly included, but most schools introduce it anyway.

In high school (grade 10–12), there is 1 hour of computer science per week for one year. This continues on the middle school content. A more rigorous computer science course is available as an elective in some schools. This course consists of three hours per week. There is an optional final examination in computer science.

There is ongoing work on revising the curriculum to make it more unified [100]. In the proposal computer science is a compulsory subject in primary schools (grades 1–6, 1 hour a week for 6 years), middle schools (grades 7–9, 1 hour a week for two years), and high schools (grade 10, 1 hour a week). There will also be an elective computer science course in high schools (grades 11–12, 3 hours a week for two years) and high school students may graduate in computer science by taking the final examination.

The unified aims are: Understanding and analysis of problems; Programing and problem solving by using computers and other digital devices; Using computers, digital devices, and computer networks; Developing social competences; and Observing law and security principles and regulations.

To support teachers there is a computer science education standards for teacher preparation, which are similar to the ISTE standards (http://www.iste.org). There is also a certification procedure, which evaluates the teachers preparation for effective and successful teaching of computer science.

J. USA

The education system in the United States is highly decentralized. Each state ad school district may have their own curricula. At the same time there is an extensive push on the national level on introducing computer science for all (https://www.whitehouse.gov/blog/2016/01/30/computer-science-all). This is a $4 billion dollar initiative from President Obama.

There is a concerted effort by all the major organizations (ACM, CSTA, Code.org, CIC, and NMSI) involved in computer science education to develop a Framework for K–12 Computer Science Education (https://k12cs.org/). They are asking big questions of the computing community: "What

Country	What?	How?	Primary	Secondary
Australia	Digital Technologies	Own subject and integrated	Compulsory	Compulsory
England	Computing	Replace existing subject	Compulsory	-
Estonia	Programming (Technology & innovation)	Integrated	Compulsory	Compulsory
Finland	Programming (Digital competence)	Integrated	Compulsory	-
New Zealand	Programming and Computer Science	Own subject	-	Elective
Norway	Programming	Own subject	-	Elective
Sweden	Programming and Digital Competence	Integrated	Compulsory	Elective
South Korea	Informatics	Own subject	Compulsory	Elective
Poland	Computer Science	Own subject	Compulsory	Compulsory
USA	Computer Science	Own subject	-	Elective

Fig. 2. Overview of the different countries.

should all students know and be able to do in K-12 computer science? What does the community expect every student to learn in elementary school, in middle school, or by the time they graduate high school? And why? Underpinning this effort is our belief that computer science provides foundational learning benefiting every child." This work is about defining the basic expectations for what every student should have a chance to learn about K-12 computer science to prepare for the emerging demands of the 21st century not just to major in computer science or secure jobs as software engineers.

The framework defines 5 concepts and 7 practices. The concepts are: Devices, Networks and Communication, Data and Analysis, Algorithms and Programming, and Impacts of Computing. The practices are: Recognizing and representing computational problems, Developing and using abstractions, Creating computational artifacts, Testing and iteratively refining, Fostering an inclusive computing culture, Communicating about computing, and Collaborating around computing.

To support high school students that want to learn computer science a new advanced placement course call CS Principles has been designed (https://advancesinap.collegeboard.org/stem/computer-science-principles). The course is intended to give a broad understanding of computer science and is organized around 7 Big Ideas of Computer Science: Creativity, Abstraction, Data and information, Algorithms, Programming, Internet and Global impact. The course is currently given for the first time and the first national exam will be in May 2017.

Besides these national efforts there are many other efforts [101], [102].

IV. Analysis

The education systems vary significantly between the countries. This makes it harder to find commonalities and general lessons learned. Three interesting dimensions are: what is being introduced? how is it introduced? and at what level(s) is it being introduced? And for each level whether it is elective or compulsory.

- What: Computing, Computational thinking, Computer Science, Digital competence, and Programming.
- How: Replace existing subject, New subject of its own, Integrated in several subjects.
- Primary/Secondary education: Elective or Compulsory.

The result can be seen in Fig.2.

By analyzing the different countries we see that the trend is to introduce computing, programming and digital competencies into primary education. Some countries have already taken the step (England and Estonia), some are about to take the step (Finland, Poland and Sweden) while others are preparing for a potential introduction later (Norway and South Korea). Many countries have had it as an elective in secondary education for quite some time (Poland, South Korea and Sweden) while others have introduced it quite recently (England, Estonia, New Zealand, and the USA). The next step is to make it either elective for more students or to make it mandatory.

The general trend is to introduce computing into primary education either in multiple subjects (Sweden) or as a cross cutting theme (Finland). England already had a subject called ICT which was remade into Computing, thereby having it as its own subject but without having to create room for a new subject. This also means that the ICT teachers now become Computing teachers.

There are many vocal proponents of having computer science as its own separate subject. From a pure subject matter perspective this is likely to be the best approach. In an ideal world this would make room in the schedule for the subject and there would be highly trained specialist teachers in the subject. However, due to practical constraints both in terms of limited school hours and the lack of highly trained teachers, many countries have chosen to introduce computing as part of existing subjects. There are some good arguments for doing this, besides the practical considerations. First, computing influences all subjects due to the digitalization of our society. Second, there is a need to develop the digital competencies of the teachers and make school more modern and relevant. If computing is its own subject, it would be very easy for all the other teachers to point their fingers on the computing teachers to take care of the problem and then continue their practice as before. By forcing more teachers to include aspects of computing in their subjects it is expected that they will have to develop their digital competencies. A risk, on the other hand, is that the quality of the teaching of computing will be relatively low.

A common struggle among all the countries is pre-service and in-service training of teachers.

V. Conclusions

The interest for computing in school has never been as high as now. Many countries have or are about to introduce computing in some form into their national curricula. At the same time the education systems differs widely. This makes it harder to compare and to learn from each other. This paper gives an overview of the different approaches in 10 different countries. The general trend is to introduce computing, often in the form of computational thinking, programming och digital competencies, into the primary education while the trend in secondary education is to develop broader courses on computer science and its impact on society, which is in contrast to previously mainly having programming courses.

The studied countries either emphasize digital competencies together with programming or the broader subject of computing or computer science. Computational thinking is rarely mentioned explicitly, but the ideas are often included in some form. The most common model is to make it compulsory in primary school and elective in secondary school. A few countries have made it compulsory in both. While some countries have only introduced it in secondary school. The common challenges to all the countries are to educate and keep good teachers with relevant knowledge and skills and to develop a suitable progression paired with teaching material throughout the education system. We expect to see much more work in the coming years. We live in exciting times.

REFERENCES

[1] Informatics Europe and ACM Europe, "Informatics in education: Europe cannot afford to miss the boat," 2015. Report of the joint Informatics Europe and ACM Europe Working Group on Informatics Education. http://europe.acm.org/iereport/ACMandIEreport.pdf.

[2] White House, "Computer science for all," 2016. https://www.whitehouse.gov/blog/2016/01/30/computer-science-all.

[3] K. E. Anja Balanskat, "Computing our future. computer programming and coding. priorities, school curricula and initiatives across europe," 2015. http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0.

[4] Department for Education, "National curriculum in England: Computing programmes of study," 2013. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study.

[5] S. Grover and R. Pea, "Computational thinking in k12: A review of the state of the field," *Educational Researcher*, vol. 42, no. 1, pp. 38–43, 2013.

[6] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson, and A. Settle, "Computational thinking in K-9 education," in *Proc. ITiCSE Working Group Reports*, 2014.

[7] E. Barendsen, L. Mannila, B. Demo, N. Grgurina, C. Izu, C. Mirolo, S. Sentance, A. Settle, and G. Stupurienė, "Concepts in K-9 computer science education," in *Proc. ITiCSE Working Group Reports*, 2015.

[8] P. Hubwieser, M. N. Giannakos, M. Berges, T. Brinda, I. Diethelm, J. Magenheim, Y. Pal, J. Jackova, and E. Jasute, "A global snapshot of computer science education in K-12 schools," in *Proc. ITiCSE Working Group Reports*, 2015.

[9] P. Hubwieser, M. Armoni, and M. N. Giannakos, "How to implement rigorous computer science education in K-12 schools? Some answers and many questions," *Trans. Comput. Educ.*, vol. 15, no. 2, pp. 5:1–5:12, 2015.

[10] L. Markauskaite, N. Goodwin, D. Reid, and P. Reimann, *Proc. ISSEP*, ch. Modelling and Evaluating ICT Courses for Pre-service Teachers: What Works and How It Works? 2006.

[11] K. Falkner, R. Vivian, and N. Falkner, "The Australian digital technologies curriculum: Challenge and opportunity," in *Proc. Sixteenth Australasian Computing Education Conference*, 2014.

[12] K. Falkner, R. Vivian, and N. Falkner, "Teaching computational thinking in K-6: The CSER Digital Technologies MOOC," in *Proc. ACE*, 2015.

[13] L. Grandell, M. Peltomäki, R.-J. Back, and T. Salakoski, "Why complicate things?: Introducing programming in high school using Python," in *Proc. 8th Australasian Conference on Computing Education*, 2006.

[14] M. Clark and R. Boyle, *Proc. ISSEP*, ch. The Transition from School to University: Would Prior Study of Computing Help? 2005.

[15] M. A. C. Clark and R. D. Boyle, *Proc. ISSEP*, ch. Computer Science in English High Schools: We Lost the S, Now the C Is Going. 2006.

[16] I. Glendinning and M. Low, *Proc. ISSEP*, ch. Collaborative Initiatives for Promoting Computer Science in Secondary Schools. 2010.

[17] S. Sentance, M. Dorling, A. McNicol, and T. Crick, "Grand challenges for the uk: Upskilling teachers to teach computer science within the secondary curriculum," in *Proc. WiPSCE*, 2012.

[18] S. Sentance, M. Dorling, and A. McNicol, *Proc. ISSEP*, ch. Computer Science in Secondary Schools in the UK: Ways to Empower Teachers. 2013.

[19] S. Sentance and S. Humphreys, *Proc. ISSEP*, ch. Online vs Face-To-Face Engagement ofComputing Teachers for their Professional Development Needs. 2015.

[20] D. Adshead, C. Boisvert, D. Love, and P. Spencer, "Changing culture: Educating the next computer scientists," in *Proc. ITiCSE*, 2015.

[21] N. Smith, Y. Allsop, H. Caldwell, D. Hill, Y. Dimitriadi, and A. P. Csizmadia, "Master teachers in computing: What have we achieved?," in *Proc. WiPSCE*, 2015.

[22] T. Bell, P. Andreae, and L. Lambert, "Computer science in New Zealand high schools," in *Proc. Twelfth Australasian Conference on Computing Education*, 2010.

[23] T. Bell, H. Newton, P. Andreae, and A. Robins, "The introduction of computer science to NZ high schools: An analysis of student work," in *Proc. WiPCSE*, 2012.

[24] D. Thompson and T. Bell, "Adoption of new computer science high school standards by new zealand teachers," in *Proc. WiPSCE*, 2013.

[25] P. Haden and J. Gasson, "It takes a village to teach information technology," in *Proc. WiPCSE*, 2013.

[26] C. Duncan and T. Bell, "A pilot computer science and programming course for primary school students," in *Proc. WiPSCE*, 2015.

[27] P. Haden, J. Gasson, K. Wood, and D. Parsons, "Can you learn to teach programming in two days?," in *Proc. Australasian Computer Science Week Multiconference*, 2016.

[28] S. Yoo, Y. Yeum, Y. Kim, S. Cha, J. Kim, H. Jang, S. Choi, H. Lee, D. Kwon, H. Han, E. Shin, J. Song, J. Park, and W. Lee, *Proc. ISSEP*, ch. Development of an Integrated Informatics Curriculum for K-12 in Korea. 2006.

[29] F. Heintz, L. Mannila, K. Nygårds, P. Parnes, and B. Regnell, "Computing at school in sweden – experiences from introducing computer science within existing subjects," in *Proc. ISSEP*, 2015.

[30] E. Gurbiel, G. Hardt-Olejniczak, E. Kolczyk, H. Krupicka, and M. M. Syslo, *Proc. ISSEP*, ch. Informatics and ICT in Polish Education System. 2005.

[31] M. M. Syslo and A. B. Kwiatkowska, *Proc. ISSEP*, ch. Informatics Versus Information Technology – How Much Informatics Is Needed to Use Information Technology – A School Perspective. 2005.

[32] M. M. Sysło and A. B. Kwiatkowska, *Proc. ISSEP*, ch. The Challenging Face of Informatics Education in Poland. 2008.

[33] M. M. Sysło and A. B. Kwiatkowska, *Proc. ISSEP*, ch. Informatics for All High School Students. 2013.

[34] M. M. Sysło and A. B. Kwiatkowska, *Proc. ISSEP*, ch. Introducing a New Computer Science Curriculum for All School Levels in Poland. 2015.

[35] C. S. Hood and D. J. Hood, "Teaching programming and language concepts using legos®," *SIGCSE Bull.*, vol. 37, no. 3, pp. 19–23, 2005.

[36] L. Buechley, M. Eisenberg, and N. Elumeze, "Towards a curriculum for electronic textiles in the high school classroom," *SIGCSE Bull.*, vol. 39, no. 3, pp. 28–32, 2007.

[37] C. W. Starr, D. Bergman, and P. Zaubi, "The development and implementation of a context-based curricular framework for computer science education in high schools," *SIGCSE Bull.*, vol. 41, no. 3, pp. 283–287, 2009.

[38] A. B. Tucker, *Proc. ISSEP*, ch. K-12 Computer Science: Aspirations, Realities, and Challenges. 2010.

[39] L. Perković, A. Settle, S. Hwang, and J. Jones, "A framework for computational thinking across the curriculum," in *Proc. ITiCSE*, 2010.

[40] S. H. Rodger, M. Bashford, L. Dyck, J. Hayes, L. Liang, D. Nelson, and H. Qin, "Enhancing k-12 education with alice programming adventures," in *Proc. ITiCSE*, 2010.

[41] S. Cooper, W. Dann, D. Lewis, P. Lawhead, S. Rodger, M. Schep, and R. Stalvey, "A pre-college professional development program," in *Proc. ITiCSE*, 2011.

[42] B. B. Morrison, L. Ni, and M. Guzdial, "Adapting the disciplinary commons model for high school teachers: Improving recruitment, creating community," in *Proc. ICER*, 2012.

[43] A. Settle, B. Franke, R. Hansen, F. Spaltro, C. Jurisson, C. Rennert-May, and B. Wildeman, "Infusing computational thinking into the middle- and high-school curriculum," in *Proc. ITiCSE*, 2012.

[44] S. Grover, S. Cooper, and R. Pea, "Assessing computational learning in k-12," in *Proc. ITiCSE*, 2014.

[45] B. J. Ericson, M. Guzdial, and T. McKlin, "Preparing secondary computer science teachers through an iterative development process," in *Proc. WiPSCE*, 2014.

[46] D. W. Lewis, L. Kohne, T. Mechlinski, and M. Schmalstig, "The exploring computer science course, attendance and math achievement," in *Proc. ITiCSE*, 2015.

[47] B. Ericson, S. Moore, B. Morrison, and M. Guzdial, “Usability and usage of interactive features in an online ebook for CS teachers,” in *Proc. WiPSCE*, 2015.

[48] A. Reiter, *Proc. ISSEP*, ch. Incorporation of Informatics in Austrian Education: The Project “Computer-Education-Society” in the School Year 1984/85. 2005.

[49] P. Micheuz, *Proc. ISSEP*, ch. 20 Years of Computers and Informatics in Austria’s Secondary Academic Schools. 2005.

[50] M. Weissenböck, *Proc. ISSEP*, ch. Informatics Education at Vocational Schools and Colleges in Austria. 2005.

[51] C. Dorninger, *Proc. ISSEP*, ch. Educational Standards in School Informatics in Austria. 2005.

[52] P. Micheuz, *Proc. ISSEP*, ch. The Role of ICT and Informatics in Austria’s Secondary Academic Schools. 2005.

[53] P. Micheuz, *Proc. ISSEP*, ch. Informatics Education at Austria’s Lower Secondary Schools Between Autonomy and Standards. 2006.

[54] P. Micheuz, *Proc. ISSEP*, ch. A Competence-Oriented Approach to Basic Informatics Education in Austria. 2011.

[55] D. L. Egger, S. M. Elsenbaumer, and P. Hubwieser, “Comparing csta k-12 computer science standards with austrian curricula,” in *Proc. WiPCSE*, 2012.

[56] D. Lessner, “Information theory on Czech grammar schools: First findings,” in *Proc. WiPCSE*, 2012.

[57] M. E. Caspersen and P. Nowack, “Computational thinking and practice: A generic approach to computing in Danish high schools,” in *Proc. Fifteenth Australasian Computing Education Conference*, 2013.

[58] J. P. Gibson, “Teaching graph algorithms to children of all ages,” in *Proc. ITiCSE*, 2012.

[59] F. Tort and B. Drot-Delange, *Proc. ISSEP*, ch. Informatics in the French Secondary Curricula: Recent Moves and Perspectives. 2013.

[60] P. Hubwieser, *Proc. ISSEP*, ch. Functions, Objects and States: Teaching Informatics in Secondary Schools. 2006.

[61] T. Brinda, H. Puhlmann, and C. Schulte, “Bridging ict and cs: Educational standards for computer science in lower secondary education,” in *Proc. ITiCSE*, 2009.

[62] A. Mühling, P. Hubwieser, and T. Brinda, “Exploring teachers’ attitudes towards object oriented modelling and programming in secondary schools,” in *Proc. Sixth International Workshop on Computing Education Research*, 2010.

[63] P. Hubwieser and A. Zendler, “How teachers in different educational systems value central concepts of computer science,” in *Proc. WiPSCE*, 2012.

[64] P. Hubwieser, M. Berges, J. Magenheim, N. Schaper, K. Bröker, M. Margaritis, S. Schubert, and L. Ohrndorf, “Pedagogical content knowledge for computer science in german teacher education curricula,” in *Proc. WiPSCE*, 2013.

[65] V. Dagdilelis and S. Xinogalos, “Preparing teachers for teaching informatics: Theoretical considerations and practical implications,” in *Proc. WiPSCE*, 2012.

[66] P. Szlávi and L. Zsakó, *Proc. ISSEP*, ch. Programming Versus Application. 2006.

[67] B. Haberman, *Proc. ISSEP*, ch. Teaching Computing in Secondary Schools in a Dynamic World: Challenges and Directions. 2006.

[68] N. Ragonis and A. Oster-Levinz, *Proc. ISSEP*, ch. Pre-service Computer Science Teacher Training within the Professional Development School (PDS) Collaboration Framework. 2011.

[69] I. Zur Bargury, “A new curriculum for junior-high in computer science,” in *Proc. ITiCSE*, 2012.

[70] I. Zur-Bargury, B. Pârv, and D. Lanzberg, “A nationwide exam as a tool for improving a new curriculum,” in *Proc. ITiCSE*, 2013.

[71] G. M. Bianco and S. Tinazzi, “One step further the ACM K-12 final report: A proposal for level 1: Computer organization for K-8,” *SIGCSE Bull.*, vol. 38, no. 3, pp. 207–211, 2006.

[72] M. C. Calzarossa, P. Ciancarini, L. Mich, and N. Scarabottolo, *Proc. ISSEP*, ch. Informatics Education in Italian High Schools. 2011.

[73] C. Mirolo, *Proc. ISSEP*, ch. A Present-Day “Glass Bead Game”: A Framework for the Education of Prospective Informatics Teachers Inspired by a Reflection on the Nature of the Discipline. 2014.

[74] V. Dagienė, *Proc. ISSEP*, ch. Teaching Information Technology in General Education: Challenges and Perspectives. 2005.

[75] V. Dagienė, G. Dzemyda, and M. Sapagovas, *Proc. ISSEP*, ch. Evolution of the Cultural-Based Paradigm for Informatics Education in Secondary Schools – Two Decades of Lithuanian Experience. 2006.

[76] V. Dagienė, *Proc. ISSEP*, ch. Teaching Information Technology and Elements of Informatics in Lower Secondary Schools: Curricula, Didactic Provision and Implementation. 2008.

[77] N. Grgurina, *Proc. ISSEP*, ch. Computer Science Teacher Training at the University of Groningen. 2008.

[78] B. Martens and T. Hofkens, “Positioning computer science in Flemish K-12 education: A reflection,” in *Proc. WiPSCE*, 2013.

[79] A. A. Kuznetsov and S. A. Beshenkov, *Proc. ISSEP*, ch. Russian Educational Standards of Informatics and Informatics Technologies (ICT): Aims, Content, Perspectives. 2005.

[80] G. Serafini, *Proc. ISSEP*, ch. Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects. 2011.

[81] S. Papert, “An exploration in the space of mathematics educations,” *International Journal of Computers for Mathematical Learning*, vol. 1, no. 1, pp. 95–123, 1996.

[82] J. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[83] J. Wing, “Computational thinking: What and Why,” 2011. http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

[84] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education, 6th Ed.* 2007.

[85] P. Hubwieser, *Proc. ISSEP*, ch. The Darmstadt Model: A First Step towards a Research Framework for Computer Science Education in Schools. 2013.

[86] Australian Curriculum Assessment and Reporting Authority, “Draft australian curriculum: Technologies,” 2013. http://www.acara.edu.au/curriculum_1/learning_areas/technologies.html.

[87] Australian Curriculum Assessment and Reporting Authority, “The australian curriculum: Technologies learning area,” 2013. http://www.austrailiancurriculum.edu.au/technologies/introduction.

[88] R. Vivian, K. Falkner, and N. Falkner, “Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development,” *Research in Learning Technology*, vol. 22, no. 0, 2014.

[89] K. Falkner and R. Vivian, “A review of computer science resources for learning and teaching with K-12 computing curricula: An Australian case study,” *Computer Science Education*, vol. 25, no. 4, pp. 390–429, 2015.

[90] N. C. C. Brown, S. Sentance, T. Crick, and S. Humphreys, “Restart: The resurgence of computer science in uk schools,” *Trans. Comput. Educ.*, vol. 14, no. 2, pp. 9:1–9:22, 2014.

[91] E. Mägi, “Tiger leap program as a beginning of 21-st century education.” http://www.ut.ee/eLSEEConf/Kogumik/Magi.pdf.

[92] HITSA, “Progetiger programme 2015 – 2017,” 2015.

[93] Finnish National Board of Education, “Perusopetuksen opetussuunnitelman perusteet 2014,” 2014.

[94] Finnish National Board of Education, “Lukion opetussuunnitelman perusteet 2015,” 2015.

[95] T. Bell, P. Andreae, and A. Robins, “A case study of the introduction of computer science in nz schools,” *Trans. Comput. Educ.*, vol. 14, no. 2, pp. 10:1–10:31, 2014.

[96] M. Sahami, S. Roach, E. Cuadros-Vargas, and R. LeBlanc, “Acm/ieee-cs computer science curriculum 2013: Reviewing the ironman report,” in *Proc. SIGCSE*, 2013.

[97] T. Bell, F. Rosamond, and N. Casey, “The multivariate algorithmic revolution and beyond,” ch. Computer Science Unplugged and Related Projects in Math and Computer Science Popularization, 2012.

[98] J. Choi, S. An, and Y. Lee, “Computing education in korea—current issues and endeavors,” *Trans. Comput. Educ.*, vol. 15, no. 2, pp. 8:1–8:22, 2015.

[99] L. Rolandsson and I.-B. Skogh, “Programming in school: Look back to move forward,” *Trans. Comput. Educ.*, vol. 14, no. 2, pp. 12:1–12:25, 2014.

[100] M. Syslo and A. B. Kwiatkowska, “Introducing a new computer science curriculum for all school levels in poland,” in *Proc. ISSEP*, 2015.

[101] M. Guzdial, B. Ericson, T. Mcklin, and S. Engelman, “Georgia computes! an intervention in a us state, with formal and informal education in a policy context,” *Trans. Comput. Educ.*, vol. 14, no. 2, pp. 13:1–13:29, 2014.

[102] J. Gal-Ezer and C. Stephenson, “A tale of two countries: Successes and challenges in K-12 computer science education in israel and the united states,” *Trans. Comput. Educ.*, vol. 14, no. 2, pp. 8:1–8:18, 2014.