Educ Inf Technol (2018) 23:419–434
DOI 10.1007/s10639-017-9611-4

Coding skills as a success factor for a society

Pauliina Tuomi[1] · **Jari Multisilta**[1] ·
Petri Saarikoski[2] · **Jaakko Suominen**[2]

Received: 26 September 2016 / Accepted: 26 April 2017 / Published online: 6 May 2017

Abstract Digitalization is one of the most promising ways to increase productivity in the public sector and is needed to reform the economy by creating new innovation related jobs. The implementation of digital services requires problem solving, design skills, logical thinking, an understanding of how computers and networks operate, and programming competence. These abilities can be considered as coding skills. The aim of the study is to find and classify the different approaches and methods of promoting and learning coding skills. In addition, coding initiatives in Finland are analyzed both from both an historical and a present-day point of view. As a result, we identified three different approaches to learning coding skills: 1) in formal settings (schools within the curriculum); 2) in non-formal settings (online, after school clubs); 3) in informal events (hackathons, jams etc.). In many cases, schools are utilizing coding events and materials created by non-profit organizations, governments, or companies. Coding is also learned in after school clubs on robotics or by creating devices using cheap computing hardware such as the Raspberry Pi.

Keywords Computational thinking · Education · Computer programming · Coding · Maker culture

✉ Jari Multisilta
jari.multisilta@tut.fi

Pauliina Tuomi
pauliina.tuomi@tut.fi

Petri Saarikoski
petri.saarikoski@utu.fi

Jaakko Suominen
jaakko.suominen@utu.fi

[1] Department of Pori, Tampere University of Technology, Pohjoisranta 11A, 28101 Pori, Finland

[2] University of Turku, Pori Unit, Pohjoisranta 11A, 28101 Pori, Finland

1 Introduction

Digital literacy is an essential component of a modern education. The twenty-first-century work environment requires digital literacy and an understanding of the key concepts of informatics. The importance of competence in computer programming, or coding, in the future, and the impact on society are widely acknowledged (Lohr 2015; Gardiner 2014). According to a recent report by BurningClass (2016), a job market analytics company, demand for coding skills in the work environment is increasing. In addition, coding skills are not just for programmers. Coding jobs are also growing faster than the overall job market.

In this article, coding skills are seen as a combination of problem solving, logical thinking, computational thinking, and design skills. According to Denning (2009), computational thinking can be defined as the ability to interpret the world as algorithmically controlled conversions of inputs to outputs. In other words, it involves learning to think about, represent, and solve problems that require a combination of human cognitive power and computing capacity (CSTB 2010; Kafai and Burke 2013; Lye and Koh 2014; Sengupta et al. 2013). The authors claim that computational thinking and coding skills are also important twenty-first-century skills. These skills refer to a wide range of capabilities, such as problem solving, critical thinking, creativity, collaboration, and communication skills (Binkley et al. 2012; Niemi et al. 2014).

The Finnish Ministry of Education has outlined that information and communication technology skills, and coding in particular, will be a fundamental part of the Finnish curriculum in 2016 (Finnish National Board of Education 2016). Until today, coding was not a mandatory school subject in Finnish schools as in the United Kingdom (UK) and Estonia, for example. Coding can be seen as a skill every citizen should master, and it will be increasingly required in future job descriptions (Wilson and Moffat 2010).

The remainder of this article is organized as follows. In section two, we present the research problem and research methods. In section three, we present the results from the study and review different methods and approaches for promoting coding skills. In section four, we discuss how Finland previously promoted computer literacy and how the new national core curriculum has changed the situation. Finally, in section five, we discuss the results and propose further research topics.

2 Research problem and methods

The aim of the study is to find and classify the different approaches and methods for promoting coding skills in formal and informal settings (i.e., in and outside school environments). We use narrative literature review as the research method to present a state-of-the-art review that considers mainly the most recent research and pilot experiments (Jones 2004). A narrative literature review can also be described as an unsystematic narrative review (Green et al. 2006), and the research method can also be described as phenomenological research. The research subject is a phenomenon, "promoting coding skills in formal and informal settings," but the data sources are the websites that describe initiatives regarding the phenomenon and research papers written on the topic.

We collected and evaluated several widespread coding initiatives from several countries and organizations. We used web pages and case descriptions found using Google search as the data sources. As a special case, we analyze the coding initiatives in Finland from a historical point of view and in terms of the current situation in autumn 2016.

3 Approaches to promising coding skills

In this section, we present the results of the review and describe different approaches and examples of promoting coding skills. The list is not exhaustive, but the intention is to give an idea of what kind of activities exist and at whom they are targeted.

First, we discuss teaching and learning coding in school as a subject. Second, we discuss afterschool activities, such as computer clubs, as a means of learning coding. Coding skills can also be learned and practiced independently from school or school-related activities. Then, we present examples of non-profit and commercial resources for learning coding skills. In the last subsection, we discuss maker culture, which enables children to learn coding by making things that combine electronics, programming, robotics, textiles, and woodworking, among others.

3.1 Coding as a subject in schools

Learning programming may not be easy (Fitzgerald et al. 2005; Vainio and Sajaniemi 2007). According to Holvikivi (2010), "research in computer science education has attempted to find efficient ways to teach programming, but usually with somewhat limited success." In general, researchers agree that students experience several difficulties in studying programming (Koscianski and Bini 2009; Lahtinen et al. 2005; Webber and Possamai 2009), such as fragile knowledge, poor problem solving skills and inabilities to apply logical thinking in a systematic way. In computer science, there are also other demanding topics, such as databases and networks. Learning these topics requires abstract thinking (Holvikivi 2010). However, computer science education can be improved by developing digital learning tools, creating new pedagogical methods, and revising the curriculum, for example (Holvikivi 2010). Schools have a responsibility to support curiosity and enquiry learning in an environment that encourages students to learn.

An interesting question is whether all children should learn coding skills in primary or secondary school, or if they should be taught only to those who aim to be programmers. We claim that coding skills are basic skills for everyone who lives in the information society. However, countries have adopted different policies for school systems in relation to learning computer programming and coding skills. For example, Estonia teaches programming to every student starting at age seven. Most countries who teach computer programming have a specific coding subject. Few countries have integrated coding as a cross curricular element to other subjects.

Educators agree that students learning programming have to develop a new way of thinking, which is more than just the syntax and semantics of a programming language (Holvikivi 2010). The most important distinction to emphasize is the difference between teaching a programming language, such as writing code and using syntax, and teaching a particular subject that develops an understanding of how computers, computer programs, and the Internet work. The aim should be that all children understand the processes and protocols that underlie today's digital technologies. This is the core of what is understood to mean coding skills.

According to Balanskat and Engelhardt (2015), by 2020, Europe may experience a shortage of more than 800,000 professionals skilled in computing and informatics. The authors claim that coding skills can help to understand today's digitalized society and foster twenty-first-century skills, such as problem solving, creativity, and logical thinking. Many countries in Europe are refocusing their curricula on developing students' coding skills and introducing the topic in their curricula. A study conducted in October 2014 among 20 European ministries of education found that computer programming and coding are already part of the curriculum in 12 countries: Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Greece, Ireland, Italy, Lithuania, Poland, Portugal, and the UK. Seven other countries also plan to integrate the topic in curricula in the future; Finland is one. For example, in the United States, coding is reportedly taught in one in ten schools (Balanskat and Engelhardt 2015).

3.2 Afterschool clubs

Various coding skill activities take place in afterschool clubs. In these club activities, children can gain hands-on experience in coding and creativity. One way to approach coding skills is robotics. Many studies indicate that educational robot activities have a positive impact on the development of students' critical thinking, problem solving, and metacognitive skills (Atmatzidou and Demetriadis 2014, 2016) and on the learning of a programming language (Alimisis 2009; Nourbakhsh et al. 2005). Other studies show that educational robot activities promote an enjoyable mode of learning, while advancing students' motivation, collaboration, self-confidence, and creativity (Eguchi 2010; Khanlari 2013; Miller et al. 2008).

In Table 1, we present examples of coding skill initiatives that are targeted for use outside school, but that can obviously also be used in school classes. As an example, in Finland there is the Innokas network, which encourages teachers, schools, and citizens to use technology in creative and innovative ways. They have created the innovative school (ISC) model that emphasizes the development of students' learning and their learning environments, teacher professionalism, leadership, and partnerships with other institutions, such as museums, libraries, and private companies. In addition, the ISC model challenges the local community, including teachers, students, school principals, and parents, to design and adopt educational innovations (Korhonen et al. 2014).

The Innokas network offers different afterschool club activities for children and organizes training courses for teachers. The network wants to point out that the ideology behind coding skills extends beyond one subject. The Innokas network supports all members of the school community to be creative and innovative. They encourage children and adults to work together to come up with new ways to use technology in everyday school life. A school is seen "as a holistic network of learning environments" that "focuses on children who are learning the skills they need to be successful in work and play in 2020" (Korhonen et al. 2014).

Raspberry Pi is a simple and cheap credit card–sized Linux computer. It is easy to program, and one can easily build electronics to extend the device's capabilities. The

Table 1 Afterschool activities

Initiative	Country, Pricing	Target Group	Mission	Activities
Ro5an kood1	Finland, free	From age 7 to 12	To support the learning of coding in primary schools	Online gaming, museums, and libraries (4.10.2016 to 31.01.2017)
CoderDojo	Global, non-profit	From age 7 to 17	An understanding of programming languages is important, and nobody should be denied the opportunity to learn	Dojo events
Raspberry Pi's Code Clubs	UK, global	After school, from age 9 to 11	To put a Code Club in every community in the world	Club activities
School Robot Challenge 2016	UK, free	School children and students	A competition to inspire schoolchildren and students with robotics and nature	Designing own virtual robot bug and teaching it to move
FabLab@School	More than 150 FabLabs over the world, low-cost	Middle and high school students	Educational digital fabrication labs that put cutting-edge technology for design and construction into the hands of middle and high school students	Workshops: designing and making digital hardware and software projects (e.g., FabLearn)

UK based Raspberry Pi Foundation supports the use of Pi's in school and in afterschool clubs, and the foundation sees after-school programming clubs as a way to kick-start the computer science revolution (Brown et al. 2014). The use of Raspberry Pi has been successful especially in schools in UK. According to Black et al. (2013), teachers in the UK are interested in computing clubs in schools but request a strong support network from business and academics.

Another example of after-school activities is an online gaming environment produced by the Finnish broadcasting company YLE (Rosa's code, or Ro5an kood1 in Finnish) that integrates afterschool activities with activities that can be done in school. The provision of activities for learning coding skills by a broadcasting company is not new. In the 1980s, the British Broadcasting Corporation (BBC) had the BBC Computer Literacy project. The aim of the project was "to introduce interested adults to the world of computers and computing, and to provide the opportunity for viewers to learn through direct experience how to program and use a microcomputer" (BBC Continuing Education Television 1982).

A widely spread after-school activity is the FabLab@School concept. Blikstein and Krannich (2013) describe FabLab@School as an application of digital fabrication and making in education. They claim that it is a new approach to bring creativity and media into schools. Stanford University launched the FabLab@School project in 2008, and so far, there are 5 FabLab@School classrooms in California, Russia and Thailand. A typical FabLab@School classroom integrates programming with laser cutters, 3D printers and other tools, so that the students can design and make both the software and hardware parts to their things.

3.3 Non-profit and commercial organizations

Diverse events take place around coding. Usually, they can easily be combined with school activities, but they are also accessible for individual participation. There are several options to learn coding worldwide through different virtual academies, such as the Microsoft Virtual Academy or Codecademy. Some academies that usually specialize in adult training also operate commercially. In Table 2, we present some online activities for learning coding skills. Non-profit and commercial organizations provide learning activities for free, for example, as a social responsibility program, or charge a fee for participating in an online course.

3.4 Maker culture and established coding events

Raspberry Pi's and similar computers that can be extended by adding your own electronics are an important part of what is called as a Maker culture. Maker culture is a contemporary culture in which people design and make things by combining different crafts techniques, such as electronics, woodworking, design, textiles, etc. Maker culture emphasizes engineering-oriented techniques, such as electronics, robotics, and three-dimensional (3D) printing, but traditional crafts activities are included. There is a strong focus on using and learning practical skills and sharing your experiences by showing work at weekly maker events and on Internet sites. Maker fairs are social gatherings that are shows and exhibitions centered on tangible items and crafts (Morozov 2014).

Table 2 Examples of non-profit and commercial organizations that provide activities for learning coding

Initiative	Country, pricing	Target group	Mission	Activities
Codecademy	Global, free, online	Everybody	Learn to code interactively for free	Interactive courses, with projects and online coding tools.
Scratch	Global, free, online	From age 8 to 16, everybody	Create stories, games, and animations to share with others around the world	Coding tools, sharing of scratch programs
Decoded	Global, commercial	Everybody	To prepare all employees for a future where the majority of jobs will require digital know-how	Classes (UK, US, AU, NL)
FreeFormers	UK, commercial	Mainly adults	Help companies and teams drive digital transformation	Workshops, events, campaigns, and content that get people embracing digital products and skills
Rasberry PI: #YesWeCode	US, non-profit	Age 18 to 30; underrepresented minorities in tech	A national initiative to help 100,000 young women and men from underrepresented backgrounds find success in the tech sector	Coding courses and training
Picademy, Raspberry Pi Foundation	UK, non-profit	Teacher at all levels	To give educators the experience and tools they need to teach computing with confidence	Teacher training, workshops, etc.

Maker culture emphasizes informal, networked, peer-led, and shared learning motivated by fun and self-fulfillment (Niemeyer and Gerber 2015). In maker culture, the learning is participatory, as many makers share their ideas and techniques with others. At the same time, people learn through the creation of new items and share their learning experiences with others (Sharples et al. 2013). Maker culture can be described as having a social constructivist basis, with social, cultural, and historical dimensions (Vygotsky 1979).

Many teachers are concerned about students' interest in STEM subjects in school. Maker culture could be used in schools to support and engage students in science, technology, engineering, and mathematics (STEM) learning. Maker activities in schools and outside vary greatly across countries and cultures (Table 3). There are several maker events (European Maker Week, Raspberry Pi Jams) and coding events (the Hour of Code, the EU Code Week, and the Year of Code), which are not related to schools in any way. However, the schools could participate to these events and integrate the event activities to the curricula. The students could learn to see the importance of STEM subjects by solving real world problems in these events.

4 Case: Finland

4.1 Historical perspective on coding skills in Finland

The Finnish educational system was shown to be excellent in the first Programme for International Student Assessment (PISA) study in 2000. The learning outcomes of 15-year-old students in reading literacy, mathematical literacy, and science literacy ranked at the top, and the differences between schools were the smallest in the world (Niemi et al. 2014). Information and communication technology (ICT) was introduced in the Finnish curriculum in the 1980s, starting first in high schools in 1982. Computer skills were seen as important, but the number of computers in Finnish schools was not adequate to reach all students. Thus, many computer activities were carried out in computer clubs after school. Computer club activity was especially important in experimental schools in the early 1980s, although some computer courses had been available at a limited level in the late 1960s and the 1970s (Saarikoski 2006, 2011). The teaching of computer skills in schools was clearly linked to the rise of the Finnish home computer market between 1983 and 1984. The early home computer boom was also clearly linked to the rise of consumer electronics. The media image of home computers was also very positive, almost overly optimistic, at the time (Saarikoski and Reunanen 2014). The concept of computer literacy was linked to the Finnish state's policy of developing an information society. The whole concept, and especially its exposure in the media, was a counterpart to similar public discussions in Europe at the same time. However, in Finland, literacy has a very strong connection to the emergence of the welfare state after World War II. Free education provided by the state was understood as the cornerstone of Finnish society. Therefore, literacy was understood as a privilege and an obligation (Saarikoski 2011).

During the 1980s, official reports and curriculum projects emphasized that youngsters should learn the basics of this new literacy. Predominantly, this meant that schools were obligated to purchase computers and other hardware. Computer training was also

Table 3 Maker culture and larger coding events

Initiative	Country, pricing	Target group	Mission	Activities
Hour of Code	Global, free	Everybody	A one-hour introduction to computer science	Students and teachers can choose from various activities, for kindergarten and up
EU Code Week	Global, free	Everybody	Aims to bring coding and digital literacy to everybody in a fun and engaging way	Workshops, online courses, taster sessions, code clubs, hackathons, theme based-events
European Maker Week	Europe, free	Everybody	To create awareness about the importance of the maker culture to foster an education of creativity and innovation in all schools across Europe	Maker fairs, FabLabs, Makerspaces, Hackerspaces, and Hardware Startup.
Year of Code	UK, free	Everybody	To encourage people across the country to get involved in coding for the first time	Hosted events about coding
Raspberry Pi Jams	Global, often small entrance fees	Everybody	Organized by the community for people to share knowledge, learn new things, and meet other Pi enthusiasts	Hands-on event, practical workshops, technical talks, show-and-tell

arranged for teachers. There were many practical problems; for example, software support was weak, and teacher training was hastily arranged. In most cases, teachers had no alternative other than to start courses with programming exercises (mostly with the BASIC programming language). The BASIC programming language was widely criticized for having too much weight in the computer curriculum. However, the programming exercises were linked to a mathematically oriented "teaching philosophy": If computer literacy was the essential ability of the information society, then programming was its official language. This can also be explained in terms of teaching policy; computer courses were mostly provided for mathematics and physics teachers (Alajääski 1987; Saarikoski 2011).

In secondary schools, ICT was introduced as a subject in the curriculum between 1987 and 1988, when it became an optional subject. In 1994, ICT started to lose its place as an individual subject. ICT skills were seen as a larger part of other subjects, meaning it could be approached through several subjects, and thus, ICT became more cross-sectional (see Sinko and Lehtinen 1999; Vahtivuori-Hänninen and Kynäslahti 2012).

Computer hobbyists played a major part in the domestication of ICT skills. User groups, clubs, and subcultures rapidly emerged all over the country during the 1980s. Networks of these communities formed a very strong social base for the emergence of the computer culture of the 1990s. Researchers have pointed out that these communities were also important when the new wave of ICT companies emerged during the 1990s. For example, Demoscene, a computer art subculture, is often mentioned as an important background to the Finnish game industry. Although this view is a little one-sided, there is plenty of evidence that Demoscene was at the root of many notable game companies, such as Remedy, Housemarque, and DICE (Saarikoski and Suominen 2009; Reunanen et al. 2013).

4.2 Coding approaches in Finland

From the maker culture point of view, Finnish schools have always embraced the idea of making things with one's own hands. Crafts have been a subject in schools and in the curriculum for decades. However, changes in society and in labor markets have affected the use of materials and techniques in the teaching of crafts (Kansanen 2004; Pöllänen 2002; Pöllänen and Kröger 2000; Pöllänen 2009). In the 1970s and 1980s, crafts subjects included textile and technical approaches. Especially for boys, electronics paved the way for different handcraft activities based on the first computers in schools. There was a lot of self-regulated play by students at that time, in which they tinkered with the actual hardware from scratch to make it work. Various guidebooks were used a great deal, such as *Build Your Own Transistor Radio*, and later, the guidebooks handled computational topics (Saarikoski 2004). The tinkering and assembling of hardware by hand represented the same kind of handcraft ideology that can be seen today in robotics, 3D printing, and such. Maker culture and crafts as a school subject is therefore a familiar phenomenon in Finnish schools. We are now experiencing the second wave that, as before, acknowledges that coding or programming is much more than just typing the code and can include operations that require a hands-on approach. In the current National Core Curriculum 2016, one crafts subject integrates both the textile and technical approaches (Finnish National Board of Education 2016).

Over the years, the curriculum has adapted to new developments, but today, handcraft skills are still considered important, even as society is increasingly oriented toward information technology (Garber 2002; Kantola 1997; Pöllänen and Kröger 2004).

Several coding-based workshops and events have also been developed in Finland that offer individual training possibilities, outside of school activities (clubs) and teacher training (Table 4).

4.3 The 2016 national core curriculum

According to Niemi et al. (2014), the "expansive use of digital technologies in education has generated the need for fresh perspectives and approaches in the development of pedagogical methods and models." The Finnish core curriculum aims to integrate technology in all subjects in an innovative way (Niemi et al. 2014). The focus on twenty-first-century skills influenced the new core curriculum for basic education in Finland, as in many other countries (Binkley et al. 2012; Kankaanranta and Vahtivuori-Hänninen 2011; Salo et al. 2011; Vahtivuori-Hänninen et al. 2014).

The new national core curriculum for primary education in Finland (from 2016 onward) states that programming, or coding, is part of all education (Finnish National Board of Education 2016). In Finland, most lessons in grades one to six are given by each class's own teacher. Therefore, these classroom teachers must be able to understand and meaningfully use coding as a teaching and learning tool. The curriculum states, for example, that for a sixth-grade student to get a good grade in mathematics, he or she must be able to create simple programs using a visual programming environment such as Scratch (although other software is available).

Currently, teachers in Finland are planning the implementation of the new national core curriculum and are looking for the best practices for teaching coding skills at all levels of primary education. The teachers and the teacher union have requested more training in coding skills for in-service teachers. The change in the core curriculum has inspired educational publishers to produce teaching and learning materials for coding skills. For example, the Koodiaapinen (Coding ABC) is an initiative by Finnish teachers and educational researchers. The website (http://koodiaapinen.fi/en/), provides updates about the current situation in Finland and shares tips and materials for teachers to learn coding skills. The new national core curriculum crystallizes the vision of education for the future and the expertise needed in Finnish society. The ongoing curriculum process is a key factor in developing twenty-first-century skills and competences in the Finnish educational sector (Niemi et al. 2014). In addition, the new learning environments and digital learning materials were chosen to be one of the key projects in the government program.

5 Discussion

This article briefly explained coding skills as a phenomenon in society and education. It also introduced global procedures for various informal and formal coding skills events. The importance of having coding competence in the future, and the impact on society, is widely acknowledged. However, approaches to enhancing citizens' coding skills have not yet been studied adequately. It is crucial to obtain information about countries

Table 4 Examples of recent Finnish approaches to learning coding skills

Initiative	Country	Target group	Mission	Activities
Innokas network	Finland, non-profit.	Children and teachers, everybody	To guide and encourage students, teachers, school administrators, and other stakeholders to be creative and innovative in the use of available technology	Robotics workshops, teacher training
Kodarit	Finland, commercial.	From age 7 to 14	Aims to bring coding to everybody in a fun, visual, and engaging way.	Camps, courses, events
Koodioulu	Finland, free.	Age 6 to 10, coding for children; and age 10 to 15, coding for youth	To create awareness about the importance of the maker culture to foster the education of creativity and innovation in all schools across Europe.	Code clubs, camps, robotics workshops
Käsityökoulu Robotti	Finland, free.	Primary and elementary school students	To combine coding and art	Camps, clubs, courses, events
Koodaustunti.fi	Finland, free	Everybody	A one-hour introduction to computer science (Finnish version of the Hour of Code)	Students and teachers can choose from a variety of activities for kindergarten and up
Koodikirja.fi	Finland	From age 4 to 120, everybody	Mission to gather all the coding training and guidebooks in one place; offers coding competence for every age	Website that offers coding tasks (to be done individually, at school, with parents, etc.)

that teach coding as a subject in school, and from informal solutions that enhance and encourage coding activity in any form. There is a need for international research that gathers information on the different strategies, in order to come up with a larger framework for today's ICT skills, coding skills, logical thinking, programming competence, and so on.

The lack of programming and computer science in K–12 education is increasingly recognized as a serious issue throughout Europe (Dagiene et al. 2014; Guerra et al. 2012). Dagiene et al. (2014) state that although informatics has been taught as a subject in many European countries as early as in the 1970s, many of these efforts were dropped for various reasons. As a result, students graduate from secondary school with a lot of experience using computers and software, but they do not have programming skills and do not understand the underlying principles of computing and computer networks (Dagiene et al. 2014).

Coding skills are becoming more important in society and work environments. We identified the different approaches used in formal, non-formal, and informal settings to foster coding skills. Coding skills are promoted by companies as a part of their business or social responsibility programs, by organizations such as science centers, by non-profits, and by different themes, such as gender, robotics, and hackerfests.

We also covered the situation in Finland. Computer clubs were important in experimental schools in the early 1980s. ICT was introduced as a subject in the curriculum between 1987 and 1988. In 1994, ICT started to lose its place as a subject as the learning of ICT skills was integrated in other subjects. Eventually, computer programming was included in the new national core curriculum again in 2016.

Coding skills are one of the most intriguing challenges that schools, educational boards, administrations, and societies face. New research is needed in order to come up with functional guidelines for teachers, as well as students, to teach and learn coding skills. Overall, awareness has risen a great deal, and the idea of mastering coding skills is widely appreciated today. This is why there are numerous guided activities based on coding sills, which were briefly presented in this article.

In Europe, the European Union has taken serious steps to improve and support the introduction of coding in schools. However, much of the attention is on research and projects, while the actual implementation is already taking place in schools, for example, the new core curriculum in Finland in autumn 2016. Therefore, it is important that teachers are educated, guided, and supported at the practical level to meet the standards of the new coding skills offered in the curriculum.

In addition to coding skills, other twenty-first-century skills are important and will be undoubtedly required in future labor markets and wider society (Gander 2013). Internationality, team, and collaboration skills will be extremely important in future work environments. It is unlikely that coding competence alone will be sufficient, as today an employee is part of a team and must communicate with others in order to achieve the results required by an employer.

References

Alajääski, J. (1987). *Tietokoneopetus peruskoulun ala-asteella: johdanto ja käsitteitä*. Turku: Turun yliopiston kasvatustieteellinen tiedekunta.

Alimisis, D. (2009). Robotic technologies as vehicles of new ways of thinking about constructivist teaching and learning: The TERECoP project [education]. *IEEE Robotics and Automation Magazine, 16*(3), 21–21.

Atmatzidou, S., & Demetriadis, S. (2014). How to support students' computational thinking skills in educational robotics activities. In *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, Padova, Italy, 18th July 2014.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems, 75*(part b), 661–670.

Balanskat, A., & Engelhardt, K. (2015). Computing our future computer programming and coding priorities, school curricula and initiatives across Europe. European Schoolnet. Retrieved September 26, 2016, from http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf.

BBC Continuing Education Television. (1982). Computer Literacy Project. Retrieved September 26, 2016, from http://www.computinghistory.org.uk/det/7182/BBC-Computer-Literacy-Project/.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw, & E. Care (Eds.), *Assessment and teaching of 21st century skills* (pp. 17–66). Dordrecht: Springer.

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P.W., & Meagher, L.R. (2013). Making computing interesting to school students: teachers' perspectives. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, 255–260. ACM.

Blikstein, P., & Krannich, D. (2013). The makers' movement and FabLabs in education: Experiences, technologies, and research. In *Proceedings of the 12th international conference on interaction design and children*, 613–616. ACM.

Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 9.

BurningClass. (2016). Beyond point and click: The expanding demand for coding skills. Retrieved September 26, 2016, from http://burning-glass.com/wp-content/uploads/Beyond_Point_Click_final.pdf.

CSTB, Computer Science Telecommunications Board. (2010). Report of a workshop on the scope and nature of computational thinking. Washington, D.C. The National Academies Press. Retrieved March 25, 2016, from http://www.nap.edu/openbook.php?record_id=12840&page=R1.

Dagiene, V. Mannila, L. Poranen, T., Rolandsson, L., and Söderhjelm, P. (2014) Students' performance on programming-related tasks in an informatics contest in Finland, Sweden and Lithuania. In *Proceedings of the 2014 conference on innovation; Technology in computer science education, ITiCSE '14*, (pp 153–158). New York, 2014: ACM.

Denning, P. (2009). The profession of IT - beyond computational thinking. *Communications of the ACM, 52*(6), 28–30.

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (Eds.), *Proceedings of society for information technology & teacher education international conference 2010* (pp. 4006–4014). Chesapeake: Association for the Advancement of computing in education (AACE).

Finnish National Board of Education. (2016). *National core curriculum for basic education 2014*. Helsinki: Publication 2016:5.

Fitzgerald, S., Simon, B., & Thomas, L. (2005). Strategies that students use to trace code: An analysis based in grounded theory. In *Proceedings of the first international workshop on computing education research* (pp. 69–80). New York: ACM Press.

Gander, W. (2013) *Informatics education: Europe cannot afford to miss the boat*. Technical report, informatics Europe & ACM Europe working group on informatics education, 2013.

Garber, E. (2002). Craft education in Finland: Definitions, rationales, and the future. *International Journal of Art & Design Education, 21*(2), 132–145 Retrieved March 23, 2016, from http://onlinelibrary.wiley.com/doi/10.1111/1468-5949.00308/pdf.

Gardiner, B. (2014). Adding coding to the curriculum. The New York Times, 23rd March. Retrieved September 26, 2016, http://nyti.ms/1ril4PT.

Green, B. N., Johnson, C., & Adams, A. (2006). Writing narrative literature reviews for peer-reviewed journals: Secrets of the trade. *Journal of chiropractic medicine, 5*(3), 101–117.

Guerra, V., Kuhnt, B., & Blochliger, I. (2012). *Informatics at school - worldwide*. Technical report, Universität Zurich.

Holvikivi, J. (2010). Conditions for successful learning of programming skills. In N. Reynolds & M. Turcsányi-Szabó (Eds.), *Key competencies in the knowledge society*, 155–164.

Jones, K. (2004). Mission drift in qualitative research, or moving toward a systematic review of qualitative studies, moving back to a more systematic narrative review. *Qualitative Report, 9*(1), 95–112.

Kafai, Y., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan, 95*(1), 61–65.

Kankaanranta, M., & Vahtivuori-Hänninen, S. (Eds.). (2011). *Opetusteknologia koulun arjessa (educational technology in schools' everyday life II)*. University of Jyväskylä: Finnish Institute for Educational Research [in Finnish].

Kansanen, P. (2004). *Opetuksen käsitemaailma*. Juva: PS-Kustannus.

Kantola, J. (1997). Cygnaeuksen jäljillä käsityönopetuksesta teknologiseen kasvatukseen. In *Jyväskylä Studies in Education, Psychology and Social Research, 133*. Jyväskylä: Jyväskylän yliopisto.

Khanlari, A. (2013). Effects of robotics on 21st century skills. *European Scientific Journal, 9*(27), 26–36

Korhonen, T., Lavonen, J., Kukkonen, M., Sormunen, K., & Juuti, K. (2014). The innovative school as an environment for the design of educational innovations. In H. Niemi, J. Multisilta, L. Lipponen, M. Vivitsou (eds.), *Finnish Innovations and Technologies in Schools* (pp. 99–113). Sense Publishers.

Koscianski, A., & Bini, E. (2009). Tackling barriers in the learning of computer programming. In *The 9th IFIP World Conference on Computers in Education*.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. In *The proceedings of ITiCSE 2005*. New York: ACM.

Lohr, S. (2015) As tech booms, workers turn to coding for career change. The New York Times, 28th July. Retrieved September 26, 2016, from http://nyti.ms/1HYAuAX.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61.

Miller, D. P., Nourbakhsh, I. R., & Siegwart, R. (2008). Robots for education. In *Springer handbook of robotics* (pp. 1283–1301). Berlin: Springer.

Morozov, E. (2014). Making it: Pick up a spot welder and join the revolution. The new Yorker. Retrieved September 26, 2016, from http://www.newyorker.com/magazine/2014/01/13/making-it-2.

Niemeyer, D., & Gerber, R. (2015). Maker culture and Minecraft: Implications for the future of learning. Educational Media International: Taylor & Francis.

Niemi, H., Multisilta, J., Lipponen, L., & Vivitsou, M. (Eds.). (2014). *Finnish innovations and technologies in schools: A guide towards new ecosystems of learning (p. 175)*. Rotterdam: Sense Publishers.

Nourbakhsh, I. R., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., & Wilkinson, K. (2005). The robotic autonomy mobile robotics course: Robot design, curriculum design and educational assessment. *Autonomous Robots, 18*(1), 103–127.

Pöllänen, S. (2002). Yhteinen käsityö sukupuolten tasa-arvon näkökulmasta tarkasteltuna. In P. Nuutinen & E. Savolainen (eds.) 50 vuotta opettajankoulutusta Savonlinnassa (pp. 219–224). Retrieved March 25, 2016, from http://sokl.uef.fi/juhlakirja/PDFt/27Sinikk.pdf.

Pöllänen, S. (2009). Contextualising craft: Pedagogical models for craft education. *International Journal of Art & Design Education, 28*(3), 249–260.

Pöllänen, S., & Kröger, T. (2000). Käsityön erilaiset merkitykset opetuksen perustana. In J. Enkenberg, P. Väisänen & E. Savolainen (eds.) *Opettajatiedon kipinöitä: kirjoituksia pedagogiikasta* (pp. 233–253). Joensuun yliopisto. Savonlinnan opettajakoulutuslaitos. Retrieved March 25, 2016, from http://sokl.uef.fi/verkkojulkaisut/kipinat/PDFt/SPetTK2.pdf.

Pöllänen, S., & Kröger, T. (2004). Näkökulmia kokonaiseen käsityöhön. In J. Enkenberg, E. Savolainen, & P. Väisänen (eds.) *Tutkiva opettajankoulutus - Taitava opettaja* (pp 160–172). Joensuun yliopisto. Savonlinnan opettajakoulutuslaitos. Retrieved September 26, 2016, from http://sokl.uef.fi/verkkojulkaisut/tutkivaope/pdft/polla_kroger.pdf.

Reunanen, M., Heinonen, M., & Pärssinen, M. (2013). Suomalaisen peliteollisuuden valtavirtaa ja sivupolkuja. In J. Suominen, R. Koskimaa, F. Mäyrä, P. Saarikoski & O. Sotamaa (eds.), *Pelitutkimuksen vuosikirja 2013* (pp. 13–28). Retrieved September 26, 2016, from http://www.pelitutkimus.fi/vuosikirja2013/ptvk2013_03.pdf.

Saarikoski, P. (2004). *Koneen lumo: Mikrotietokoneharrastus Suomessa 1970-luvulta 1990-luvun puoliväliin*. Finland: Jyväskylän yliopisto. (in Finnish).

Saarikoski, P. (2006). Koneen ja koulun ensikohtaaminen. Suomalaisen atk-koulutuksen varhaisvaiheet peruskoulussa ja lukiossa. *Tekniikan Waiheita, 3*, 5–19.

Saarikoski, P. (2011). Computer courses in Finnish schools during 1980-1995. In *History of Nordic Computing HiNC3, 3rd IFIP WG9.7 Working Conference on History of Nordic Computing,* Stockholm 18th–20th October 2010.

Saarikoski, P., & Reunanen, M. (2014). Great northern machine wars – History and cultural impact of the rivalry between user groups in Finland from the 1980s to the 2000s. *IEEE Annals of the History of Computing, 2*.

Saarikoski, P., Suominen, J. (2009). Computer hobbyists and the gaming industry in finland. IEEE Annals of the History of Computing. Perspectives on the History of Computer Games, 31 (3).

Salo, M., Kankaanranta, M., Vähähyyppä, K., & Viik-Kajander, M. (2011). Tulevaisuuden taidot ja osaaminen. Asiantuntijoiden näkemyksiä vuonna 2020 tarvittavasta osaamisesta [Future skills and know-how]. In M. Kankaanranta & S. Vahtivuori-Hänninen (eds.), Opetusteknologia koulun arjessa II. University of Jyväskylä.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380.

Sharples, M., McAndrew, P., Weller, M., Ferguson, R., Fitz Gerald, E., Hirst, T., & Gaved, M. (2013). *Innovating pedagogy 2013: Open University innovation report 2*. Milton Keynes: The Open University.

Sinko, M., & E. Lehtinen (1999). Challenges of ICT in Finnish education. Retrieved April, 3, 2006, from http://www.sitra.fi/Julkaisut/sitra227.pdf.

Vahtivuori-Hänninen, S., & Kynäslahti, H. (2012). ICTs in a school's everyday life. In H. Niemi, A. Toom, & A. Kallioniemi (eds.). *Miracle of education: The principles and practices of teaching and learning in Finnish schools* (pp. 237–248). Sense Publishers.

Vahtivuori-Hänninen, S., Halinen, I., Niemi, H., Lavonen, J., & Lipponen, L. (2014). A new Finnish national core curriculum for basic education (2014) and technology as an integrated tool for learning. In H. Niemi, J. Multisilta, L. Lipponen, & M. Vivitsou (Eds.), *Finnish innovations and technologies in schools: A guide towards new ecosystems of learning* (pp. 21–32). Rotterdam: Sense Publishers.

Vainio, V., & Sajaniemi, J. (2007). Factors in novice programmers' poor tracing skills. In *In The Proceedings of ITiCSE 2007*. New York: ACM.

Vygotsky, L. S. (1979). Consciousness as a problem in the psychology of behaviour. *Soviet Psychology, 17*, 3–35.

Webber, C.G., & Possamai, R. (2009). An immune-based approach to evaluate programming learning. In *The 9th IFIP World Conference on Computers in Education*.

Wilson, A., & Moffat, D. C. (2010). Evaluating scratch to introduce younger schoolchildren to programming. In *The Proceedings of the Psychology of Programming Interest Group Workshop,* September, Madrid/Espanha.