Tensions in specifying computing curricula for K-12: Towards a principled approach for objectives

13 authors, including:

Some of the authors of this publication are also working on these related projects:

CaReWood View project

Towards a Meaningful Instrumental Music Education. Methods, Perspectives, and Challenges View project

Mary E. Webb*, Tim Bell, Niki Davis, Yaacov J. Katz, Andrew Fluck, Maciej M. Sysło, Ivan Kalaš, Margaret Cox, Charoula Angeli, Joyce Malyn-Smith, Torsten Brinda, Peter Micheuz, and Andrej Brodnik

Tensions in specifying computing curricula for K-12: Towards a principled approach for objectives

Abstract: In this article we examine key issues and tensions for developing and specifying Computing-related elements of curricula design, particularly the role of Computer Science in the curriculum. The article is based on a series of discussions and analyses of curriculum design across various countries with different approaches and traditions of Computing in the curriculum.

Keywords: School curriculum, Computer Science, Curriculum objectives, Digital Citizenship

ACM CCS: Social and professional topics → Professional topics → Computing education → Model curricula, Computational thinking, K-12 education

1 Introduction

This article stems from a series of discussions since 2012 led by members of the "Task Force on Curriculum" of the Education Committee of the International Federation of Information Processing (IFIP). The Task Force was set up in response to concerns originating from many different countries about the Computing Curriculum for K-12 being in need of reform because Computer Science has been neglected [13, 26, 34]. In our early discussions we reached consensus regarding the following key ideas in relation to the Computing Curriculum [31, 32]:

1) Computer Science and digital literacy are complementary – both are needed in the school curriculum;
2) any curriculum specification needs room for flexibility in interpretation; and
3) the terminology in relation to Computing is important but complex and variable.

Crucially several areas of controversy were also identified and have remained contentious in subsequent discussions [30]. Key contentious issues are whether or not **all** students should study Computer Science, from what age and for which purpose: Digital Citizenship, creating informed consumers, developing professionals with Computing expertise and understanding of Computing to fulfil the needs of the Computing industry per se and for other industry and enterprises? Our aim in this article is to examine why these issues are contentious, how they have played out in curriculum design across different countries, what were the considerations and/or tensions involved and how and to what extent they have been resolved. In this endeavour, this article draws on three main sources: 1) outcomes of the Task Force discussions; 2) analysis of literature on curriculum design, both in relation to Computing Education as well as general curriculum theory and 3) analysis of curriculum design considerations across different countries.

2 Background and terminology

It is now several years since the crisis in the curriculum for Computer Science was identified in many Western coun-

*Corresponding author: Mary E. Webb, King's College London, London, UK, e-mail: mary.webb@kcl.ac.uk
Tim Bell, Niki Davis, University of Canterbury, Christchurch, New Zealand, e-mails: tim.bell@canterbury.ac.nz, niki.davis@canterbury.ac.nz
Yaacov J. Katz, Bar-Ilan University, Ramat-Gan, Israel and Michlala – Jerusalem Academic College, Jerusalem, Israel, e-mail: Yaacov.Katz@biu.ac.il
Andrew Fluck, University of Tasmania, Launceston, Australia, e-mail: Andrew.Fluck@utas.edu.au
Maciej M. Sysło, UMK Toruń, Toruń and University of Wrocław, Wrocław, Poland, e-mail: syslo@ii.uni.wroc.pl
Ivan Kalaš, Comenius University, Bratislava, Slovakia, e-mail: Ivan.Kalas@fmph.uniba.sk
Margaret Cox, King's College London, London, UK, e-mail: MJ.Cox@kcl.ac.uk
Charoula Angeli, University of Cyprus, Nicosia, Cyprus, e-mail: cangeli@ucy.ac.cy
Joyce Malyn-Smith, Education Development Center, Waltham, Massachusetts, USA, e-mail: jmsmith@edc.org
Torsten Brinda, Universität Duisburg-Essen, Duisburg, Germany, e-mail: torsten.brinda@uni-due.de

Peter Micheuz, Alpen-Adria-University of Klagenfurt, Klagenfurt am Wörthersee, Austria, e-mail: peter.micheuz@AON.AT
Andrej Brodnik, University of Ljubljana, Ljubljana, Slovenia, e-mail: andrej.brodnik@fri.uni-lj.si

Mother tongues in local language	Art	Geography	History	Mathematics	Religion/ethics/citizenship	Science	Sport/PE	Computer Science/Informatics Computing/Digital Technologies

Computational Thinking

Cyberwellness
Digital Citizenship
Digital Literacy
Media Literacy
Operational skills

Computational Thinking

Computational Thinking

Computing and associated theories studied as a separate subject

IT/ICT - the applications and connectivity used by students to enhance learning in all subjects

Information Technology - the hardware and operating systems operated by people in educational contexts.
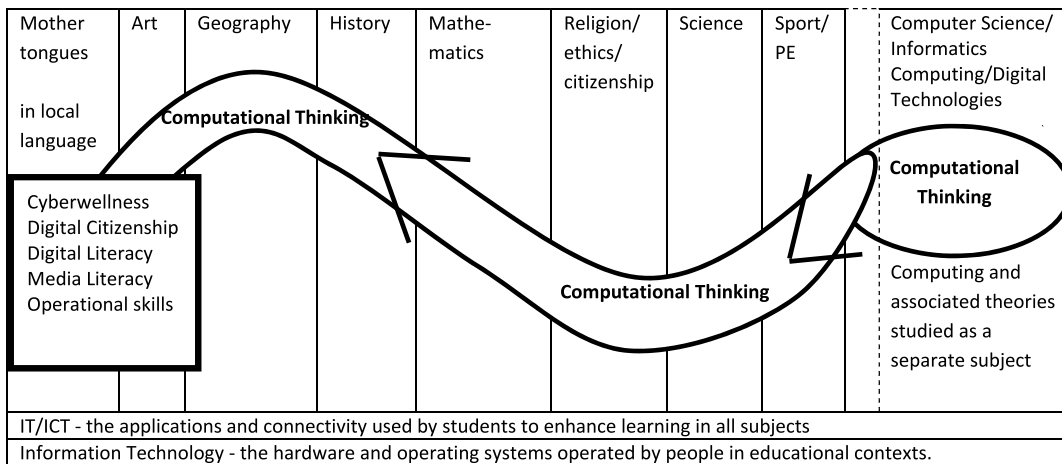
Figure 1: Computing-related terms in relation to the conventional school curriculum.

tries. Most have responded with curriculum initiatives, but the issues have by no means been fully resolved. For example, the UK government, during 2016, undertook an inquiry into digital skills in the UK resulting in a report entitled 'Digital Skills Crisis' [11]. This report identified a skills gap throughout education and training as well as in industry and business and identified considerable challenges for schools, teachers and teacher education in providing students with the digital skills needed now and for a future not yet imagined. In that report 'digital skills' referred to the broad range of skills needed to make use of digital technologies and the report referred to the importance of the revised UK's Computing Curriculum as well as to the need to develop digital skills across the whole curriculum.

As mentioned earlier terminology is important and problematic. Therefore, in Figure 1 we present a diagram outlining Computing-related terms in relation to the school curriculum that we agreed on at EDUsummIT[1] 2015 [7, 32]. The figure illustrates that there is a curriculum subject based on an academic discipline and given various names in different countries and contexts, which in this article we will refer to as Computer Science. Other aspects of Computing, especially Computational Thinking, typically exist within this specialist subject but also across the whole curriculum where computational thinking can enhance traditional subjects [1]. In Figure 1 Computational Thinking is shown penetrating across the curriculum as a wavy block to indicate variability in its implementation and some uncertainty and its interpretation. Computational Thinking is an approach to problem solving

conceived by Seymour Papert [20] and given prominence as a key element in curricula more recently by Jeanette Wing [35]. Computational Thinking involves analysing and solving problems and presenting solutions in such a way that they can be implemented on computers and therefore draws on fundamental concepts of Computer Science. The diagram illustrates the broad nature of the curriculum elements that are related specifically to Computer Science. Furthermore, analysis of global curricula and frameworks for the skills needed for participation in a digital world show a consensus regarding the importance of digital literacy but also include skills in the related aspects of communication, collaboration, problem solving, critical thinking, creativity and productivity [29]. Therefore, technological developments over the last approximately 50 years and the accelerating rate of change in the 21st Century have created a need for significant new areas of knowledge and skills that can lead to tensions when incorporating them into existing, often crowded, curricula.

In this article, we do not intend to discuss the content of Computer Science in depth but in order to consider rationale and objectives, it is necessary to indicate the nature of the discipline and the broad content area of Computer Science. A working definition of Computer Science as a curriculum subject reflecting recent consensus and current practice [7], is "the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information" [33]. Analysis of several recent curriculum specifications [30] showed a consensus over the broad conceptual content of Computer Science curricula: programming/algorithms,

---

1 EDUsummIT is a biennial International Summit Meeting of educators, policymakers, practitioners, industry representatives and researchers who meet to analyse and make recommendations regarding current priorities in relation to IT and education.

Table 1: Entitlement and starting age for Computer Science across six countries [adapted and updated from 30].

Theme	Australia	Israel	NZ	Poland	Slovakia	UK
Entitlement- who is the COMPUTER SCIENCE curriculum for?	New curriculum for all	All must learn Computer Science and technology literacy incorporating computation-al thinking	High school subject for seniors from 2011; all children in primary schools from 2018	High and middle school subject for 20 years. All from 7–11 and an advanced option from 8–12.	All from elementary school upwards	All from elementary school upwards
Starting age for COMPUTER SCIENCE	From the first year of school (about 5 years old)	Elementary School	From 2018 from the first year of school	7 years old	8 years old	5 years old

data representation, digital infrastructure, digital applications, human factors and related ethical issues. There was less agreement regarding the importance of more general intellectual practices and social competences such as cooperation, collaboration and communication. A tension was identified between the desirability of incorporating such social skills and intellectual practices into the curriculum and the difficulties of assessing the skills [30].

Arguments for the inclusion of Computer Science in the school curriculum are compelling, as evidenced by many countries adopting it as a mandatory part of their curriculum. However, as a relatively new subject Computer Science has to compete with other more well-established curriculum priorities. Rationales for incorporating Computer Science into the curriculum were summarised as economic, social and cultural and while all of these rationale are crucial, how they are operationalised in a curriculum design depends on the particular country and context [7, 30]. It may also be related to disciplinary traditions that can impact curriculum interpretation and implementation overtly and covertly. In particular, the engineering discipline perceives less need for evidence before action when compared with the scientific disciplines from which Computer Science has emerged [25]. The economic rationale rests both on the need to produce Computer Scientists and for Computer Science-enabled professionals to support innovation and development in all fields and enterprises. The social rationale emphasises the value of active creators and producers with power to lead, create and innovate within society. The cultural rationale rests on enabling people to be drivers of cultural change rather than having change imposed upon them as a result of technological developments. The rationales outlined here present a strong case for the importance of Computer Science in the curriculum, but questions as to whether the subject should be compulsory, and at what age it should be introduced, also require consideration.

3 Issue: Should all students study Computer Science?

As explained above the ability to create and produce with technologies provides individuals with power in society. Therefore, access to the "powerful knowledge" [37] of Computer Science is an issue of individual entitlement for students. Analysis of curriculum developments [14, 30] (See Table 1) showed that the importance given to this entitlement varied across countries. In the UK the Royal Society report emphasised entitlement and opportunity for all students [26], as does the change in the national curriculum for New Zealand that begins in 2018 [17]. Likewise, Poland and Slovakia emphasised the importance of opportunity and the goal of motivating all students to use computational thinking and to engage in solving problems as well as to prepare students to consider Computing and related fields as disciplines of their future study and professional career. Similarly, the Australian curriculum "Digital Technologies" rationale [2] emphasised entitlement as well as economic needs. While Australia, New Zealand, Poland, Slovakia and the UK have taken the firm view that Computer Science within the curriculum is for all, Israel has opted for a segregated model dependent on students' capabilities. It is noteworthy however that in Israel, basic "Computing and technology literacy" incorporating computational thinking is an entitlement for all, based on the importance of developing problem-solving and analytical thinking skills. This "literacy" goes beyond the typical digital literacy that focuses only on *using* computers rather than problem-solving and computational thinking.

Thus, we see in these countries, curriculum developments emphasising both entitlement and economic considerations. It is likely that economic considerations are a major driver of curriculum developments in many

countries, owing to a shortage and perceived increasing need for Computing expertise. In addition to the well-recognised importance of technological developments for economic prosperity [see for example 21], cyberattacks, especially the Ransomware incidents in 2017, have revealed the vulnerability of organisations that do not deploy sufficient expertise to manage the security of their systems.

If we set aside the entitlement issue then, as some have argued previously, economic priorities might be met by encouraging students, who have shown their capabilities in mathematics and other science subjects, to move into Computer Science at a later stage in education. However, there are significant arguments for starting Computer Science at a young age, as we shall discuss next.

4 Issue: What age to introduce students to Computer Science?

As many in the Computing profession have argued, programming is difficult, and it takes many years to learn [see 22, for a review]. While programming is only one element of Computer Science, it is generally regarded as essential in any introductory course. Furthermore, while Computing professionals do not necessarily write code themselves, they need to understand essentials of programming when making decisions about development and management of computer systems etc. Decisions about the early introduction of Computer Science in the UK and Poland were partly based on a view from school teachers and computer scientists in higher education that aspects of Computer Science, especially programming, require gradual acquisition and development over many years. There is also evidence that exposing students to key Computational Thinking concepts before 12 years old is not only possible but important for developing their self-efficacy, lack of which can disadvantage girls particularly and thus affect gender diversity in University courses and the IT industry [5]. Moreover, while developing Computational Thinking is not dependent on programming, the ability for students to evaluate their thinking by implementing and running a program is a powerful learning tool. Therefore Computational Thinking should be developed in Computer Science learning and applied across other curriculum subjects [7]. Furthermore, if learners are never introduced to Computer Science as a disciplinary area and to the knowledgebase and approaches that Computing academics and professionals use, then they are unlikely to be able to determine whether this discipline is for them, and this is widely regarded as a

key factor in the lack of gender diversity in the Computing industry [6]. The age of introduction of Computer Science into the curriculum is therefore an entitlement and equity issue. In recent years the increased availability of programming environments and other applications and tools, including robots that can be easily programmed, designed to support young learners in learning programming has supported the introduction of Computer Science at a young age e. g. seven years old [5]. Thus, students can learn through hands-on experience and gradually begin to link theoretical concepts to their developing practical problem-solving capabilities. In this way curriculum design can overcome a major difficulty of reconciling theory and practice [23].

Thus far we have argued that there are economic, social and cultural rationale for the inclusion of Computer Science in the curriculum and that introducing all students to Computer Science from a young age, typically about 7–8 years old, is not only possible, with the resources now available, but is an entitlement and equity issue as well as being important for economic prosperity. However, as we indicated earlier, Computer Science is the academic discipline that we have argued is essential but is not the only aspect of new technological developments that must be considered when designing a curriculum. There are questions of "Digital Citizenship" and decisions about the overall balance of the curriculum in relation to Computing and the extent to which its purpose is to create Digital Citizens, informed consumers and/or Computing-enabled professionals and what the curriculum requirements of such purposes might be.

5 Issue: Digital Citizenship

While there is controversy about the extent to which all students need to study Computer Science, few would disagree that to participate fully in a world driven by technology, Digital Citizenship is essential for all. However precisely what Digital Citizenship entails is less clear. A study of educators' understanding of Digital Citizenship in the US, [10] suggested that there is a need to clarify the aims of Digital Citizenship education and that such education needs to start at an early age. According to Searson, Hansen et al., Digital Citizenship is a broad term that incorporates aspects of online behaviour and "cyber wellness" in a changing world of technology [24]. A concept analysis, based on a survey of 500 university students, identified Internet political activism, technical skills, local/global awareness, critical perspective and networking agency as

the main concepts of Digital Citizenship [3]. In the context of school education, the emphasis for Digital Citizenship may currently be more limited. For example the International Society for Technology in Education (ISTE) includes Digital Citizenship as one of seven standards that students need to thrive in a constantly evolving technological landscape and provides the following four indicators: "1) Students cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world; 2) Students engage in positive, safe, legal and ethical behaviour when using technology, including social interactions online or when using networked devices; 3) Students demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property and 4) Students manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online" [12]. These indicators emphasise responsive, appropriate and safe online behaviours rather than the creative and innovative behaviours that are indicated by the social and cultural rationales for Computer Science in the curriculum discussed earlier. A key issue is to define the underlying knowledge base of Computer Science concepts and crucial skills needed to support Digital Citizenship. The skills required for the behaviours indicated in Digital Citizenship depend on, but may extend beyond, the digital literacy mentioned earlier. For instance, The National Curriculum for Computing in England defines being digitally literate as students being 'able to use, and express themselves and develop their ideas through information and communication technology at a level suitable for the future workplace and as active participants in a digital world.' [4, P. 1]. What is needed to support the social and cultural rationales and to enable the behaviours associated with the Digital Citizenship, is the development of understanding of underlying principles and concepts of Computer Science alongside the development of practical skills and behaviours. Through such a curriculum, underpinned by Computer Science understanding, learners can not only behave appropriately as Digital Citizens in the current Internet-based world but also evaluate new technologies for their potential to support their own learning and creativity. Furthermore, they will be enabled to evaluate the wider importance for society of technological developments and to take their place in a changing world. The need to keep up with changing technologies, and the knowledge that is needed to be able to respond to a future that is not yet imagined, is a major issue for future-proofing the curriculum as we will discuss next.

6 Issue: Future proofing the curriculum

There are concepts and principles of Computer Science that have remained relatively stable over many years. For example, the von Neumann architecture first described in 1945 is still the basic architecture underlying most computer systems. It is the improved processing power derived from increasing complexity combined with reduced component size as predicted by Moore's Law [18] of exponential improvement in cost and complexity that has led to massive development in the range and capability of computer systems. Thus, some knowledge in Computer Science curricula can remain relatively stable while examples of applications change to make the curriculum practically useful and relevant. However, there are aspects of Computer Science knowledge and principles that are developing rapidly and therefore Fluck et al [7] argued that a curriculum must find ways of accommodating new developments in Computer Science knowledge. Furthermore, in future, changes in the underlying fundamental knowledge base of Computer Science may be more rapid. For example, some scientists are now claiming that Moore's Law no longer holds, and therefore change will be driven by different technological approaches and architectures [see for example 28]. One important area of development is Quantum Computing, in which the fundamental components, qubits, can exist in a probabilistic superposition of several states simultaneously, thus calling into question the relevance of a curriculum based on understanding principles of binary states and procedural algorithms [7]. Such developments suggest the need for a curriculum that has some relatively stable components but can also respond quickly to change.

The Digital Citizenship model discussed earlier is almost entirely based on dealing with the Internet and its applications including particularly social media. Obviously, the Internet, together with the World Wide Web and social media applications, has had a massive impact across most aspects of life, and is arguably the most significant technological development to which education has had to respond. But other technological developments may lead to equally significant or even more profound change. For example, major changes are occurring in the area of Artificial Intelligence (AI) which provides tremendous opportunity for growth and development, while also causing concern. Specific AI-supported products are already important in many aspects of life including everyday applications such as the digital assistants in smartphones; and have changed the world of work by creating opportunities for humans

to partner with machines in solving intractable problems such as accurately diagnosing diseases and reducing hospital acquired infections. Advances in specific AI systems are giving rise to automation, a concern of large numbers of the working public. In the near future, specific AI systems are predicted to be able to perform a majority of tasks currently undertaken by humans, more quickly and cost effectively [15]. This increase in automation poses a challenge for managing the potential changes to employment patterns that are likely to follow.

Artificial General Intelligence (AGI), which aims to create machines with human levels of intelligence across a broad spectrum of capability, is more controversial. AGI and the risks that it poses by becoming capable of developing itself incrementally into a "super intelligence" have been brought to the fore in recent years through eminent figures such as Bill Gates and Stephen Hawking expressing their concerns. For example, in 2014, Stephen Hawking stated "The rise of powerful AI will be either the best or the worst thing ever to happen to humanity. We do not know which yet" [9]. Many scientists now believe that AGI is possible but disagreements remain regarding how and when it will be achieved [15]. The World Economic Forum, in its global risks assessment in 2017, acknowledged the threats posed by AGI and examined the implications for governance [36].

There is therefore a strong argument for school curricula to include consideration of AI and to identify and incorporate relevant knowledge into curriculum and such knowledge may involve radical departure from existing curricula. For example, quantum Computing, discussed earlier, is one of the approaches that holds promise for supporting rapid developments in AI [19]. In order to ensure that a curriculum has the flexibility to respond to new developments, Fluck et al. proposed that a substantial part of the final year of the curriculum should be reserved for emergent technologies and updated annually [8].

7 Issue: Delivering the curriculum for Computer Science

Thus far we have focused on arguing for inclusion of curriculum content for Computer Science based on curriculum rationale and purposes but there are considerations associated with the delivery of curriculum that can affect decisions regarding development and implementation [30]. These factors include availability of resources, elucidation of a research-informed pedagogy especially for early years learning [16], but most particularly the avail-

ability of suitably qualified and experienced teachers and the commitment to provide ongoing professional development to enable these teachers to respond to the need for frequent updating of the curriculum in response to technological change. Teacher professional development was identified as a major challenge for countries that have reintroduced Computer Science into their curricula [32].

8 Conclusion: From rationale to objectives for computing in the curriculum

We have argued that access to Computer Science in the curriculum is an entitlement for all students from an early age. Furthermore, curriculum design should be guided by economic, social and cultural rationales when incorporating Computer Science into the curriculum. Recent concerns about a lack of Computer Science in the curriculum were based predominantly on an economic rationale [13, 26, 27, 34]. The Royal Society's [27] vision does mention empowering people in a changing world. It argues that science and mathematics are central to addressing many global challenges as well as being important for maximising young people's chances of success personally and professionally thus: "In science and mathematics there is a fortunate coincidence between the intellectual and cultural needs of the individual and the economic needs of the nation" [27, P. 17]. In line with the vision of the Royal Society, we believe Computer Science has a role, not only underpinning other sciences and Digital Citizenship, but also a transformational role through discovery and innovation. Therefore, Computer Science should be incorporated in the curriculum from approximately age 7 and then throughout the rest of compulsory schooling. Furthermore, we have argued that although some Computer Science based innovations, such as AI, provide world-changing potential benefits, they can also present plausible threats to our future. Therefore, it is crucial to develop a curriculum that incorporates Computer Science as an academic discipline that not only identifies new ways to innovate but also provides both tools and techniques for other sciences and understanding to support Digital Citizenship. Computer Science is vital to empower young people with the knowledge and skills, so they can choose their places in society, and then make critical decisions about the future of the world, as we know it.

In order to support the development of curricula incorporating Computer Science, a number of challenges

remain. As we have discussed, there is already evidence that young students, of 7–8 years old can start to develop understanding of important Computer Science concepts. Therefore, identifying trajectories in the development of these concepts and devising effective pedagogical approaches which make use of the tools available are important current research challenges. Furthermore, in addition to developing Computer Science concepts to support the subject per se, it is necessary to define the underlying knowledge base of Computer Science concepts and crucial skills needed to support Digital Citizenship. This quest is ongoing as the technology continues to develop and to impact on all aspects of our lives.

References

1. C. Angeli, J. Voogt, J. Malyn-Smith, M. E. Webb, A. Fluck, M. Cox, and J. Zagami, A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Education Technology and Society, (2016). 19(3): p. 47–57.
2. Australian Curriculum Assessment and Reporting Authority (ACARA). Australian Curriculum. 2017; Available from: https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/.
3. M. Choi, M. Glassman, and D. Cristol, What it means to be a citizen in the internet age: Development of a reliable and valid digital citizenship scale. Computers & Education, (2017). 107: p. 100–112.
4. Department for Education, National curriculum in England: computing programmes of study. (2013): London, England.
5. C. Duncan, T. Bell, and S. Tanimoto. Should your 8-year-old learn coding? In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WIPSCE 2015). (2014). Berlin, Germany: New York, ACM.
6. A. Fisher and J. Margolis, Unlocking the clubhouse: Women in computing. (2003), Cambridge, Massachusetts: MIT press.
7. A. Fluck, M. E. Webb, M. Cox, C. Angeli, J. Malyn-Smith, J. Voogt, and J. Zagami, Arguing for Computer Science in the School Curriculum. Education Technology and Society, (2016). 19(3): p. 38–46.
8. A. Fluck, Timely updates for Computer Science curricula in World Conference on Computers in Education (WCCE) (2017): Dublin.
9. S. Hawking. Speech at the launch of the Leverhim Centre for the Future of Intelligence. 2016; Available from: http://www.cam.ac.uk/research/news/the-best-or-worst-thing-to-happen-to-humanity-stephen-hawking-launches-centre-for-the-future-ofv.
10. R. Hollandsworth, J. Donovan, and M. Welch, Digital Citizenship: You Can't Go Home Again. TechTrends, (2017): p. 1–7.
11. House of Commons Science and Technology Committee, Digital skills crisis. (2016).
12. ISTE. ISTE standards for students. 2017; Available from: https://www.iste.org/standards/standards/for-students.
13. Joint Informatics Europe & ACM Europe Working Group on Informatics Education, Informatics education: Europe cannot afford to miss the boat: Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. (2013).
14. M. Kabátová, I. Kalaš, and M. Tomcsányiová, Programming in Slovak Primary Schools. Olympiads in Informatics, (2016). 10: p. 125–159.
15. S. Makridakis, The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms. Futures, (2017). 90: p. 46–60.
16. A. Manches and L. Plowman, Computing education in children's early years: A call for debate. British Journal of Educational Technology, (2017). 48(1): p. 191–201.
17. Ministry of Education New Zealand, Digital Technologies | Hangarau Matihiko. Draft for consultation. (2017).
18. G. E. Moore, Cramming more components onto integrated circuits. Electronics Magazine, (1965). 38(8).
19. V. Moret-Bonillo, Can artificial intelligence benefit from quantum computing? Progress in Artificial Intelligence, (2015). 3(2): p. 89–105.
20. S. Papert, An exploration in the space of mathematics educations. International Journal of Computers for Mathematical Learning, (1996). 1(1): p. 95–123.
21. J. Pollex and A. Lenschow, Surrendering to growth? The European Union's goals for research and technology in the Horizon 2020 framework. Journal of Cleaner Production, (2016).
22. A. Robins, J. Rountree, and N. Rountree, Learning and Teaching Programming: A Review and Discussion. Computer Science Education, (2003). 13(2): p. 137–172.
23. J. J. Schwab, The Practical: Arts of Eclectic. The School Review, (1971). 79(4): p. 493–542.
24. M. Searson, M. Hancock, N. Soheil, and G. Shepherd, Digital citizenship within global contexts. Education and Information Technologies, (2015). 20(4): p. 729–741.
25. M. Tedre and M. Apiola, Three computing traditions in school computing education. In Improving computer science education, D. Kadijevich, C. Angeli, and C. Schulte, Editors. (2013), Routledge: London. p. 100–106.
26. The Royal Society, Shut down or restart? The way forward for computing in UK schools. (2012), The Royal Society: London.
27. The Royal Society, Vision for science and mathematics education, in The Royal Society Science Policy Centre report 01/14. (2014).
28. T. N. Theis and H. S. P. Wong, The End of Moore's Law: A New Beginning for Information Technology. Computing in Science & Engineering, (2017). 19(2): p. 41–50.
29. J. Voogt, O. Erstad, C. Dede, and P. Mishra, Challenges to learning and schooling in the digital networked world of the 21st century. Journal of Computer Assisted Learning, (2013). 29(5): p. 403–413.
30. M. Webb, E. N. Davis, T. Bell, Y. J. Katz, N. Reynolds, D. P. Chambers, and M. M. Sysło, Computer science in K-12 school curricula of the 2lst century: Why, what and when? Education and Information Technologies, (2017). 22(2): p. 445–468.
31. M. E. Webb. Considerations for the design of Computing curricula. In KEYCIT 2014 – Key Competencies in Informatics and ICT. (2014). University of Potsdam, Germany: Commentarii informaticae didacticae (CID) | 7.

32. M. E. Webb, A. Fluck, M. Cox, C. Angeli-Valanides, J. Malyn-Smith, J. Voogt, and J. Zagami, Thematic Working Group 9: Curriculum – Advancing Understanding of the Roles of Computer Science/Informatics in the Curriculum, in EDUsummIT 2015 Summary Report: Technology Advance Quality Learning for All, K.-W. Lai, Editor. (2015): Bangkok, Thailand. p. 60–69.
33. Wikipedia. Computer Science. 2018; Available from: https://en.wikipedia.org/wiki/Computer_science.
34. C. Wilson, L. A. Sudol, C. Stephenson, and M. Stehlik, Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age. (2010), Association for Computing Machinery (ACM), Computer Science Teachers Association (CSTA).
35. J. Wing, Computational thinking. Communications of the ACM, (2006). 49(3): p. 33–36.
36. World Economic Forum Global Risks Report 2017 (2017).
37. M. Young, Overcoming the crisis in curriculum theory: a knowledgebased approach. Journal of Curriculum Studies, (2013). 45(2): p. 101–118.

Bionotes

Mary Webb
King's College London, London, UK
**mary.webb@kcl.ac.uk**

Dr Mary Webb is Reader in Information Technology in Education at King's College London. As Subject Director for the Computer Science PGCE, Mary has trained secondary teachers of Computing for many years and before that she taught Science and Computing in primary and secondary schools. Mary's research has examined computer science education, pedagogy and formative assessment and uses of IT for learning and has resulted in over 150 publications. Mary is on the Executive of the International Federation of Information processing (IFIP) Education Committee and leads the Task Force on Computer Science in the curriculum.

Tim Bell
University of Canterbury, Christchurch, New Zealand
tim.bell@canterbury.ac.nz

Prof Tim Bell is in the Department of Computer Science and Software Engineering at the University of Canterbury in Christchurch, New Zealand. His main current research interest is computer science education; in the past he has been also worked on computers and music, and data compression. His "Computer Science Unplugged" project is widely used internationally, and its books and videos have been translated into over 20 languages. In 2018 he received the ACM SIGCSE Outstanding Contribution to Computer Science Education award. Recently he has been actively involved in the design and deployment of new computer science curriculum in New Zealand schools.
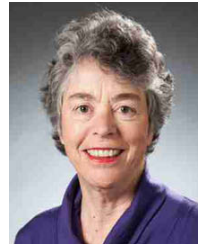
Niki Davis
University of Canterbury, Christchurch, New Zealand
niki.davis@canterbury.ac.nz

Niki Davis is Distinguished Professor of e-Learning and Director of the e-Learning Lab in the University of Canterbury, New Zealand where she also coordinates UC's postgraduate programmes in 'e-Learning and Digital Technologies' in education. This includes ongoing collaboration with Tim Bell and his team in Computer Science. She has an international reputation for technology in teacher education recognised with awards from IFIP and SITE. Current research includes a leading role for the digital world in NZ National Science Challenge 'A Better Start, e tipu e rea'. Her most recent book is Digital technologies and change in education (Routledge, 2018).

Yaacov J. Katz
Bar-Ilan University, Ramat-Gan, Israel and Michlala – Jerusalem Academic College, Jerusalem, Israel
Yaacov.Katz@biu.ac.il

Prof Yaacov J Katz is Professor Emeritus at the School of Education, Bar-Ilan University in Israel and now serves as President of Michlalah – Jerusalem Academic College. He specializes in religious education & values, affective education, ICT use in education and social attitudes in education. Prof Katz served as Head of the School of Education at Bar-Ilan University and as Chief Pedagogic Officer of the Israel Ministry of Education where he was responsible for all subject matter taught in Israeli state schools.

Andrew Fluck
University of Tasmania, Launceston,
Australia
Andrew.Fluck@utas.edu.au

Dr. Fluck has taught science, mathematics and computing in Nigeria, England and Australia. He works as a senior lecturer in information technology education in the Faculty of Education at the University of Tasmania. Dr. Fluck's funded research investigates the transformative use of computers to teach integral calculus and quantum mechanics in primary schools; and eExams, where students take their own computers into the exam hall. He is the chair of Working Group 3.3 (research into educational applications of information technologies) for IFIP/UNESCO. Andrew is also an avid longbow archer and continental archery judge. See more details at http://Andrew.Fluck.id.au.

Maciej Sysło
UMK Toruń, Toruń and University of
Wrocław, Wrocław, Poland
syslo@ii.uni.wroc.pl

Maciej M. Syslo, mathematician (over 150 publications) and computer scientist – academic and school teacher, instructor at in-service courses for teachers, author of informatics and ICT curricula, educational software, school textbooks and guidebooks for teachers; member of several national committees on education, Polish representative to IFIP TC3; recipient of national and international awards: Steinhaus (Poland, 1986), Car (Poland, 2010), Best Practices in Education Award (Informatics Europe, 2013), IFIP Outstanding Service Award (2014), and grants: Mombusho (Japan, 1974–1976), Humboldt (Germany, 1982–1984), Fulbright (USA, 1996–1997).

Ivan Kalaš
Comenius University, Bratislava, Slovakia
Ivan.Kalas@fmph.uniba.sk

Ivan Kalaš is a professor of Informatics education (Computing) at Comenius University, Bratislava. His professional interests include development of constructionist educational interfaces for learning programming, and research in the field of the impact of digital technologies on learning. Ivan is a co-author of several programming environments for children, including Comenius Logo, Imagine Logo, Thomas the Clown and RNA (Revelation Natural Arts) adopted by thousands of schools, home and abroad. Between 2014 and 2016 he was a lead developer of the UCL ScratchMaths programming interventions for English primary school pupils aged 9 to 11. Currently he leads a development of a novel approach to learning programming as a tool for exploring the concepts around us for pupils aged 8 to 9.

Margaret Cox
King's College London, London, UK
MJ.Cox@kcl.ac.uk

Margaret Cox is Professor of IT in Education at King's College London, and Professorial Fellow at the University of Melbourne. She has been researching and developing IT in education since 1969, also pioneering the use of haptics in Health care education, which includes directing the award winning hapTEL project. She is the current President of the National Conference of University Professors; joint Editor of the Section (1st and 2nd Edition) Researching IT in Education in the International Handbook entitled the use of IT in primary and Secondary Education which formed the foundation of the Edusummits. As a founder member of the Edusummits she has been a Programme Committee member on since 2009.
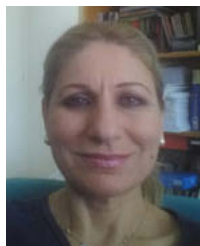
Charoula Angeli
University of Cyprus, Nicosia, Cyprus
cangeli@ucy.ac.cy

Dr. Charoula Angeli is Professor of Instructional Technology at the University of Cyprus. She has undergraduate and graduate studies at Indiana University-Bloomington, USA (BS in Computer Science, MS in Computer Science, and Ph.D. in Instructional Systems Technology). She also pursued post-doctoral studies at Learning Research and Development Center (LRCDC), University of Pittsburgh, USA. Her research interests include the utilization of educational technologies in K-12, the design of computer-enhanced curricula, joint cognitive systems, computational thinking, technological pedagogical content knowledge, educational software design, teacher training, teaching methodology, online learning, and the design of learning environments for the development of thinking skills.

Joyce Malyn-Smith
Education Development Center, Waltham,
Massachusetts, USA
jmsmith@edc.org

Joyce Malyn-Smith (Ed.D.) leads a body of work at EDC, Inc. (USA) that focuses on how people develop technology skills and knowledge in and out of school, then translate those into a productive and rewarding career. She works with industry to define new/emerging skill sets and dispositions needed to work successfully at the human-technology frontier; and works with educators to integrate those skills and dispositions into curricula K-20. Currently she leads several large-scale, innovative projects focused on the future of work, STEM education, computer science, computational thinking and data/big data.

Torsten Brinda
Universität Duisburg-Essen, Duisburg,
Germany
torsten.brinda@uni-due.de

Prof. Dr. Torsten Brinda studied Computer Science at the University of Dortmund (Diploma 1998). He started his scientific career in the Computing Education Research Groups at the universities of Dortmund (until 2002) and Siegen (until 2005), where he 2004 received his doctoral degree. From 2005 to 2012 he worked as a full professor for computing education research at the University of Erlangen-Nuremberg, since then in the same position at the University of Duisburg-Essen. He is an active member of ACM SIGCSE, IFIP TC3 and the German Informatics association (GI), where he is the current chairman of the technical committee for "Computing Education Research/Computing and Education".

Peter Micheuz
Alpen-Adria-University of Klagenfurt,
Klagenfurt am Wörthersee, Austria
peter.micheuz@AON.AT

Peter Micheuz, born in 1955, is since 1979 an Austrian teacher at the Alpen-Adria-Gymnasium Völkermarkt and since 2000 in charge of teachers' education for Informatics at the Alpen-Adria-University Klagenfurt. He is author of textbooks and (inter)national papers, editor of conference proceedings and engaged in regional and national IT projects. He is member of the ECDL board in Austria and since 2015 vice-chair of the IFIP TC3 Working Group 3.1.

Andrej Brodnik
University of Ljubljana, Ljubljana, Slovenia
andrej.brodnik@fri.uni-lj.si

Andrej Brodnik is affiliated with the University of Primorska and the University of Ljubljana. His main research interests include succinct data structures, Computer Science education, and ubiquitous systems. Prof. Brodnik is a member of IEEE and ACM. He is also a winner of the National award for exceptional achievements in higher education and of a number of other national and international rewards. He is author of over 150 scientific papers and conference contributions, and 6 international patents. Google Scholar list over 1,500 citations of his work.