EDUsummIT Thematic Working Group 9 and IFIP TC3 Curriculum Task Force Curriculum - Advancing Understanding of the Roles of Computer Science/Informatics in the Curriculum

Summary Report and Action Agenda (updated July 2016)

Mary Webb, King's College London, UK
Andrew Fluck, University of Tasmania, Australia
Margaret Cox, King's College London, UK
Charoula Angeli, University of Cyprus, Cyprus
Joyce Malyn-Smith, Education Development Center, USA
Joke Voogt, University of Amsterdam, Netherlands
Jason Zagami, Griffith University, Australia

Introduction

This short report, written by EDUsummIT[1] Thematic Working Group 9, results from a series of discussions, over several years, led by the IFIP TC3 Task Force: "Curriculum - deeper understanding of roles of CS/Informatics", as well as from discussions before, during and after EDUsummIT 2015. The report was originally published in 2015 (Webb et al., 2015), and was updated in July 2016, incorporating elements from discussions in the IFIP TC3 Task Force and a symposium at the IFIP TC3 SaITE Conference (Webb et al., 2016).

At EDUsummIT 2015 it was argued that the major rationales for including Computer Science as a subject in the K-I2[2] curriculum are economic, social and cultural. The economic rationale rests not only on the need for a country to produce computer scientists to sustain a competitive edge in a world driven by technology but also on the requirement for Computer Science-enabled professionals in all industries to support innovation and development. The social rationale emphasises the value in society of active creators and producers rather than passive consumers of technology. Such capability provides people with power to lead, create and innovate within society and therefore is also an issue of entitlement to "powerful knowledge" (Young, 2013) giving individuals opportunities to choose their role in society. The cultural rationale rests on enabling people to be drivers of cultural change rather than having change imposed by technological developments. These arguments are presented in detail in Fluck at al. (2016).

In this brief report we explain firstly the background to recent curriculum changes and the global context in which they are taking place. Then we explain the issues and challenges for establishing and maintaining the roles for Computer Science in curricula for K-12. Next we present solutions and recommendations for policy makers, educators, industrial partners and researchers and finally we outline our actions for taking forward these solutions.

[1] EDUsummIT is a global community of researchers, policy-makers and educators committed to supporting the effective integration of Information Technology (IT) in education by promoting active dissemination and use of research.
[2] K-I2 refers to kindergarten through to Grade 12 (aged 4-19 years).

Background and Context

Previous EDUsummITs examined broadly the skills needed in a world driven by technology. EDUsummIT 2011 identified the importance of new competencies for 21st-century learning, including digital literacy, in the curriculum (Voogt, Erstad, Dede, & Mishra, 2013). EDUsummIT 2013 examined computational thinking as a critical set of thinking skills of equal importance to literacy and numeracy in the education of all (Voogt, Fisser, Good, Mishra, & Yadav, 2015). This importance of computational thinking was highlighted by Wing (2006), especially in the context of Computer Science, although the concept is generally attributed to Papert (1980, 1996) who examined computational thinking in relation to learning mathematics and other subjects. Subsequently Barr and Stephenson's (2011) work led to an operational definition published by the Computer Science Teachers Association (CSTA) incorporating the following core characteristics:

- "Formulating problems in a way that enables us to use a computer and other tools to help solve them;
- Logically organizing and analyzing data;
- Representing data through abstractions such as models and simulations;
- Automating solutions through algorithmic thinking (a series of ordered steps);
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;
- Generalizing and transferring this problem solving process to a wide variety of problems." (http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf).

The recent curriculum developments that formed the impetus for the current report include a review of the information and communications technology (ICT) curriculum in the UK (The Royal Society, 2012) that identified a need for major reform and similar calls in the United States (Wilson, Sudol, Stephenson, & Stehlik, 2010; Cuny, 2012), Canada (http://cacsaic.org/HowAlbertaGotCS), throughout Europe (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013), Australia (http://www.australiancurriculum.edu.au/technologies/digital-technologies/rationale) and New Zealand (Bell, Andreae & Robins, 2012). These initiatives emphasise refocusing Computing/ICT education to incorporate Computer Science as the underlying subject discipline. In these countries where reform was recommended, curricula that had previously contained Computer Science had become weakened or diverted by other priorities. In other countries, such as Israel and Cyprus, Computer Science has been retained since its emergence in the 1980s. However, even in countries that have maintained Computer Science, rationales for its presence in curricula and curricula themselves vary (Hazzan, Gal-Ezer, & Blum, 2008; Webb, Davis, Reynolds, & Sysło, 2015) and are worthy of scrutiny.

The economic imperative is a strong driver behind the recent refocusing on Computer Science in the USA and Europe and focuses on the benefits of people being able to apply technological solutions and entrepreneurial talent to solve problems and promote sustainable development (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013; The Royal Society, 2012; Wilson et al., 2010). At the same time many educators see the importance of people developing understanding of the capabilities and potential of technologies and being able to engage in "computational thinking" and other forms of thinking promoted by studying Computer Science, such as systems thinking, in order to support learning and create informed and responsible citizens. This new thinking and understanding is not the digital literacy whose importance is already well-established but a set of skills, understanding and thinking that can be developed by engaging with and understanding Computer Science; understanding how computers work and designing and creating computer-based solutions, including through programming.

Key Challenges and Issues

Table 1 summarises the main challenges and their solutions identified at EDUsummIT 2015 and indicates recommendations for policymakers (P), educators (E), industrial partners (I) and researchers (R). Evidence of these challenges and justification for solutions is outlined below.

Table 1 Challenges and Solutions for Advancing Understanding of the Roles of Computer Science/Informatics in the Curriculum

Challenge	Solution/Recommendation to P, E, I, R
1. Lack of clear understanding (outside the field of Computer Science) of Computer Science/Informatics as an academic discipline.	(a) Adopt a globally agreed statement of Computer Science/Informatics as a discipline in its own right (P, I, R, E). (b) Articulate the nature, importance and relevance of Computer Science/Informatics to society and education (P, I, R & E).
2. A need for Computer Science/Informatics as a distinct subject in school curricula is controversial and poorly understood.	Disseminate and communicate a clear rationale to different stakeholders about the need to have Computer Science/Informatics as a distinct subject in school curricula (P, I, R & E).
3. Computational thinking, a core component of Computer Science/Informatics, is considered to be important in 21st century skills, but due to its complexity, it is difficult to implement in schools.	Promote computational thinking through the means of a Computer Science/Informatics curriculum, which aims at making computational thinking commonplace (P, R & E).
4. The development of Computer Science/Informatics school curricula is impeded by insufficient empirical evidence of student learning in order to support content definition and sequencing.	Design Computer Science/Informatics curricula based on a content analysis, and then continue to research students' learning difficulties as well as the effects of different pedagogical approaches. (E & R).
5. Previous ICT curricula deliveries poorly prepared students for Computer Science/Informatics in further/higher education or professional employment.	Facilitate better smart partnerships between education systems and industry/professional associations. (E & I)
6. Integrating Computer Science/Informatics across other subjects in school curricula has been ineffective.	Identify clear learning outcomes, assessments and standards for Computer Science/Informatics. (E, I, P & R)

Challenge	Solution/Recommendation to P, E, I, R
7. Teacher professional development in a newly introduced Computer Science/Informatics subject is a challenge in quality and quantity for many countries.	**a) Encourage more Computer Science/Informatics graduates to become teachers. (P, I & E)** **b) Add a Computer Science/Informatics specialisation to pre-service training for primary school teachers. (P & I)** **c) Make Computer Science/Informatics professional learning a requirement for periodic teacher re-accreditation/licensing. (P)** **d) Schools need resource allocations to free teachers to undertake the professional learning and preparation for a new Computer Science/Informatics subject. (P)**
8) Identifying and allocating the additional resources for teaching Computer Science/Informatics is a challenge.	**(a) Some of Computer Science/Informatics can be taught without computers. But computers especially mobile devices can enhance the learning experience. (P, E, I & R)** **(b) Teacher training needs to provide skills in using the available resources in the most efficient way. (E)** **(c) Identify, and if not available, commission teaching support materials in mother-tongue language especially for younger students (P,E,I).**
Key: Policy maker (P), Educator (E), Industry partners (I), Researcher (R)	

1. Lack of understanding of Computer Science/Informatics as an academic discipline

The nature of Computer Science as an academic discipline is relatively invariant across cultures, and a common definition[3] is widely accepted. However, developments in Computer Science and consequent technological innovations are rapid, making it difficult for people to understand their importance and relevance. This rapid change also mitigates against clear understanding of the importance of Computer Science in the K-12 curriculum.

2. A need for Computer Science/Informatics as a distinct subject

The major rationales for including Computer Science in the K-12 curriculum as a distinct subject were outlined earlier in this paper as were the characteristics of computational thinking. Our recommendation for promoting computational thinking through learning Computer Science is based on the understanding that computational thinking is integral to problem-solving approaches in Computer Science (Wing, 2006). Furthermore, working with concepts from Computer Science and programming, which is not usually taught in other curriculum subjects, provide very practical ways to engage with computational thinking even for young children, for example, programming a robotic

---

[3] This is a definition of Computer Science, which is widely accepted: "the scientific and practical approach to computation and its applications. It is 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."[Wikipedia] It seeks to answer the following questions: What is information? What is computation? How does computation expand what we know? How does computation limit what we can know? (Denning, 2007).
Informatics, which is a term used widely in education in Europe, incorporates Computer Science but is broader and encompasses the entire set of scientific concepts that make information technology possible (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013).

toy (such as the "Beebot") to follow forward/left/right movements to reach a destination. Computational thinking and other skills developed in Computer Science, including programming, can then be applied to other curriculum areas (see, for example, Barr & Stephenson, 2011).

3. *Computational thinking is difficult to implement in schools*

As argued above, computational thinking involves developing ways of solving problems, designing systems, and understanding human behaviour that draws on concepts fundamental to computer science. However, large-scale studies of computer practices in schools (e.g. Voogt & Knezek, 2008; Plomp, Anderson, Law & Quale, 2009) and earlier reviews (Cox & Abbott, 2004) have shown that within the current school curriculum teachers neither have sufficient time nor expertise to implement computational thinking. The addition of Computer Science as a distinct subject would lead to expert teachers in the field and the curriculum time for computational thinking to become an integral part of every child's education. In Angeli et al. (2016), we elucidate an approach to developing computational thinking skills for K-6 that can motivate students and develop usable skills and confidence through tackling real-world problems.

4. *The development of Computer Science/Informatics school curriculum is impeded by insufficient empirical evidence*

Defining a Computer Science appropriate curriculum structure and sequencing is challenging because there is less evidence of how students develop understanding of Computer Science compared with other subjects. However, there are epistemological considerations and constraints which can guide curriculum design (Winch, 2013; Young, 2013). Furthermore, lessons from curriculum theory and from experiences of curriculum design in other subjects suggest that we need to live with uncertainty and to accept the need for a dynamic and continually renegotiated curriculum (Webb, 2014) and at the same time continue to research how students develop their understanding. In conjunction with specification of curriculum structure and sequencing there is a need for clear learning outcomes, assessments and standards for Computer Science/Informatics throughout K-12. Such clear specification should help to alleviate the problems of confusion that have contributed to the failure of integrated solutions for incorporating Computer Science/Informatics into the curriculum (Wilson, Sudol, Stephenson, & Stehlik, 2010). Therefore, given the additional challenges of teacher professional development, we recommend that in primary education, while it may well be integrated across the curriculum in its delivery, Computer Science is identified as a subject area with specialist support while in secondary schools Computer Science is taught as a separate subject by specialist teachers.

5. *Previous ICT curricula deliveries poorly prepared students for Computer Science/Informatics in further and higher education or professional employment*

Most countries have had national programmes and policies for many years regarding the incorporation and teaching of ICT in schools (Plomp et al., 2009). However, for many years, national curricula in many countries have mainly focused on teaching basic computer skills such as word-processing, using email, drawing a graphics program, communication using Email and Chat and searching for information using the Internet (ibid.) and not the teaching of computational thinking which is a very important 21st century skill. A Computer Science curriculum would have this subject as its core and ensure all school pupils developed competence in it.

6. *Integrating Computer Science/Informatics across other subjects in the school curriculum has been ineffective*

In spite of enormous growth in the number of computers in schools and widespread access to the Internet, an international study of policies and practices of using ICT across the curriculum in 37 countries (Plomp et al., 2009) has concluded that the integration of ICT use into other school subjects such as science and mathematics, let alone the teaching of Computer Science, has been spasmodic and in many schools non-existent. For example, reporting on the United States in 2009, Andersen & Dexter stated that, “Many of America’s 54 million elementary and secondary school students remain largely unaffected by existing technological infrastructure. Large numbers of teachers and students rarely touch a computer” (Andersen & Dexter, 2009, p. 707).

7. *Teachers’ professional development in a newly introduced Computer Science/Informatics subject is a challenge in quality and quantity for many countries*

With the shift from ICT across the curriculum incorporated by teachers of a range of different curriculum subjects to a requirement for schools to deliver a CS/Informatics curriculum there has been an urgent need to train existing and new teachers in this subject. To date, in most developed and developing countries, there are insufficient specialist computing teachers to teach this curriculum in all schools (Webb, 2014). In Angeli et al. (2016), we examine the new technological pedagogical content knowledge that teachers need to teach computational thinking, as one element of Computer Science, as well as the challenges for professional development and how they can be addressed.

8. *Identifying and allocating the additional resources for teaching Computer Science/Informatics is a challenge*

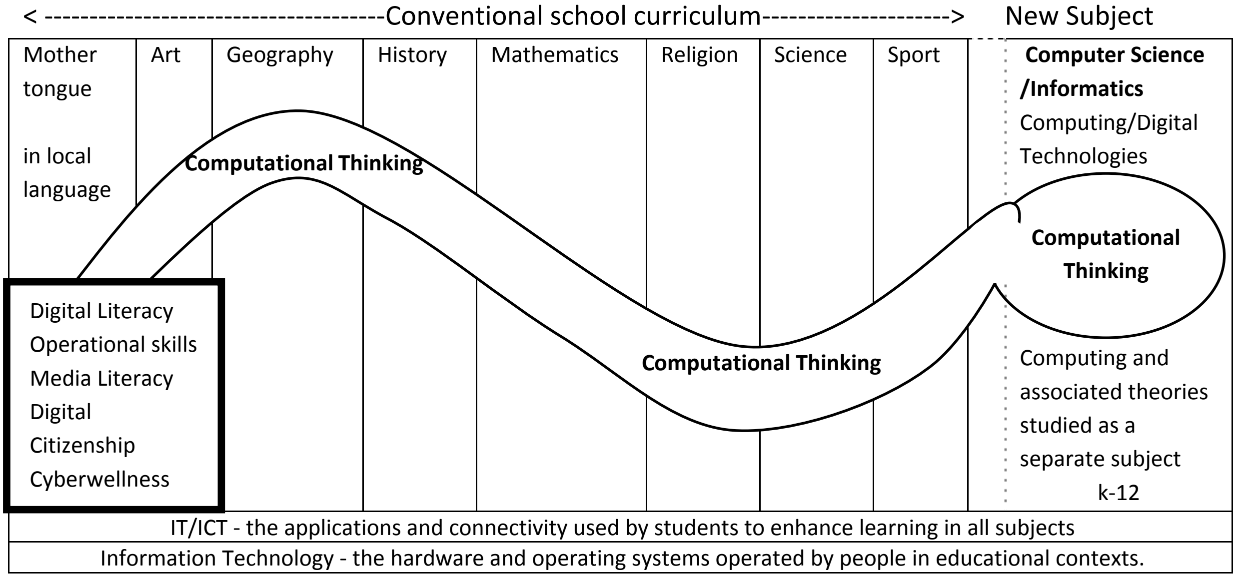
The challenge of providing appropriate resources for teaching Computer Science obviously varies between countries (Plomp et al., 2009) but there are ways of teaching many elements of Computer Science without using computers, e.g. the "unplugged" approach (Bell & Newton, 2013) which can be engaging for students. However, there are possibilities for new software resources for teaching Computer Science enabling improved or accelerated learning of difficult concepts. This is particularly evident in learning programming, which is generally regarded as difficult, but developments of tools for visualising programming, for example, can improve learning provided teachers understand their pedagogical significance (see Ben-Ari, 2013, for a review). Thus it was agreed that resource provision should not be a barrier to introducing Computer Science into the curriculum but making available a good range of resources and ensuring teachers are able to use them efficiently should be an ultimate goal.

While the order of challenges shown in Table 1 represents a logical progression for considering curriculum rationale and design, the order of priority and difficulty will vary across contexts. For example, currently teacher professional development is a major challenge for those countries that are introducing or re-introducing Computer Science.

Critical for global communications about these challenges is the issue of varying terminology used in different countries. Figure 1 offers a mapping of such diversity and the position of the elements of computing-related terms in relation to the conventional school curriculum.

Further issues that were addressed by other EDUsummIT working groups and are relevant to understanding of the roles of Computer Science/Informatics in the curriculum include: 1) the interrelationships between curricula, pedagogy and assessment; 2) the relationship between formal and informal learning; 3) the need to take account of informal learning in assessment. Another issue, at a finer grained level of detail, that concerns policy makers and educators, is the choice of programming languages for learning. We agreed that it is the problem-solving processes and design of algorithms that should take precedence over the learning of specific programming languages. However, it was also agreed that program implementation, testing and debugging is essential and the choice of language affects learning opportunities and pedagogy. Furthermore the predominance of English as the basis for programming languages mirrors the wider tension between schooling in a child’s own language and the importance of accessing international material.

Figure 1 Computing-related terms in relation to the conventional school curriculum



Action Plan

The EDUsummIT working group has completed the following elements of the action agenda from EDUsummIT 2015.

1. Elaborated 2 detailed research papers to address challenges and issues raised in this paper on a)"Arguing for Computer Science in the School Curriculum" (Fluck et al., 2016) and b) "Defining Pedagogical Content Knowledge needed for primary and secondary teachers to teach Computer Science" (Angeli et al., 2016)
2. Disseminated outcomes at various conferences including WERA at AERA 2016 and SaITE 2016

The working group and IFIP TC3 Curriculum Task Force will:

3. Continue to disseminate outcomes at various conferences including WCCE 2017 and the next EDUsummIT 2017
4. Develop a UNESCO Policy Paper on Advancing Understanding of the Role of Computer Science in the Curriculum
5. Create a further research paper to address challenges and issues raised in this paper on the "Challenges for specifying structure and sequence in the Computer Science curriculum: the interrelations between resource issues and pedagogical approaches"
6. Inform national governments of the findings and recommendations outlined in this report

Future work of the IFIP TC3 Curriculum Task Force will include:

7. Addressing challenges and issues raised in this report by a) examining how Computer Science has been incorporated into the curriculum in various contexts and b) examining frameworks that support the development of Computer Science curricula in various contexts
8. Examining how computational thinking can be developed and consolidated within Computer Science curricula and wider curricula contexts
9. Examining the relationship between computational thinking, digital and other literacies and Computer Science.

References

Anderson, R.E. & Dexter, S. National Policies and Practices on ICT in Education: United States. In Plomp, T., R.E. Anderson, Law, N. & Quale, A. (2009). Eds. *Cross-National Information and Communication Technology: Policies and Practices in Education*. Information Age Publishing. Charlotte. USA.

Angeli, C., Voogt, Malyn-Smith J., Webb M. E., Fluck A., Cox M. and Zagami J. (2016). "The Design of the Computer Science Curriculum and the Knowledge that Teachers Need to Teach Computer Science in K-12 " *Education Technology and Society 19(3): 47-57*.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the Computer Science education community? . *ACM Inroads, 2*(1), 48-54.

Bell, T., Andreae, P., & Robins, A. (2012) Computer Science in NZ High Schools: The First Year of the New Standards. In L. A. S. King, D. R. Musicant, T. Camp, & P. Tymann (Eds.), *Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, NC, USA (pp. 343–348).* New York: ACM.

Bell, T., & Newton, H. (2013). Unplugging Computer Science. In D. M. Kadijevich, C. Angeli & C. Schulte (Eds.), *Improving computer science education* (pp. 66-81). New York, London: Routledge.

Ben-Ari, M. (2013). Visualisation of programming. In D. M. Kadijevich, C. Angeli & C. Schulte (Eds.), *Improving Computer Science education* (pp. 52-65). New York, London: Routledge.

Cox, M. J. and Abbott, C. (Eds.), (2004) *ICT and attainment – a review of the research literature. Full Report*. Becta Coventry/DfES London January 2004.

Cuny, J. (2012). Transforming High School computing: A Call to Action. *ACM Inroads*, 3(2), 32-36.

Denning, P.J. (2007) Computing is a Natural Science. *ACM Communications 50(7)13-18*.

Fluck, A., M. E. Webb, M. Cox, C. Angeli, J. Malyn-Smith, J. Voogt and J. Zagami (2016). "Arguing for Computer Science in the School Curriculum." *Education Technology and Society 19(3): 38-46*.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). A model for high school computer science education: the four key elements that make it! *SIGCSE Bull., 40*(1), 281-285. doi: 10.1145/1352322.1352233.

Joint Informatics Europe & ACM Europe Working Group on Informatics Education. (2013). *Informatics education: Europe cannot afford to miss the boat: Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education.*

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123.

Plomp, T., R.E. Anderson, Law, N. & Quale, A. (2009). Eds. *Cross-National Information and Communication Technology: Polices and Practices in Education*. Information Age Publishing. Charlotte. USA.

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools.* London: The Royal Society.

Voogt, J. & Knezek, G. (Eds.) *International Handbook of Information Technology in Primary and Secondary Education*. Berlin, Heidelberg, New York. Springer.

Voogt, J., Erstad, O., Dede, C., & Mishra, P. (2013). Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning*, 29(5), 403-413.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 1-14.

Webb, M. E., A. Fluck, M. Cox, C. Angeli-Valanides, J. Malyn-Smith, J. Voogt and J. Zagami (2015). Thematic Working Group 9: Curriculum - Advancing Understanding of the Roles of Computer Science/Informatics in the Curriculum. EDUsummIT 2015 Summary Report: Technology Advance Quality Learning for All. K.-W. Lai. Bangkok, Thailand: 60-69, available at: http://www.curtin.edu.au/edusummit/local/docs/edusummit2015-ebook.pdf.
Webb, M. E., C. Angeli, Y. Chtouki, M. Syslo, M. Lokar, P. Micheuz, A. Fluck, M. Cox, J. Malyn-Smith, J. Voogt, J. Zagami, N. Mori and A. Brodnik (2016). Symposium: Roles of CS/ Informatics in the curriculum – why, what, how*? Stakeholders and Information Technology in Education ('SAITE') Conference*. Universidade do Minho, Guimarães, Portugal.
Webb, M. E. (2014, July 1st – 4th, 2014). Considerations for the design of Computing curricula. Paper presented at the K*EYCIT 2014 – Key Competencies in Informatics and ICT*, University of Potsdam, Germany.
Webb, M., Davis, N., Katz, Y. J., Reynolds, N., & Sysło, M. M. (2015 July 1 - 3, 2015). *Towards deeper understanding of the roles of CS/ Informatics in the curriculum*, Vilnius University, Lithuania.
Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. Association for Computing Machinery (ACM), Computer Science Teachers Association (CSTA).
Winch, C. (2013). Curriculum Design and Epistemic Ascent. *Journal of Philosophy of Education*, 47(1), 128-146.
Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-36.
Young, M. (2013). Overcoming the crisis in curriculum theory: a knowledge-based approach. *Journal of Curriculum Studies, 45(2), 101-118*.

Acknowledgements

Thanks to:

1. Contributors to the EDUsummIT 2015 discussion paper on which this paper drew: Yousra Chtouki, Al Akhawayn University in Ifrane, Morocco; Niki Davis & Tim Bell University of Canterbury, Christchurch, New Zealand; Yaacov J Katz, il Michlala - Jerusalem Academic College and Bar-Ilan University, Israel; Nicholas Reynolds & Dianne P. Chambers, The University of Melbourne, Australia; and Maciej M. Sysło, University of Wrocław and UMK, Toruń, Poland.
2. Other members of the IFIP TC3 Curriculum Task Force for comments and suggestions at various stages and especially those whose work informed the pre-EDUsummIT discussions.
3. Those who contributed papers to the symposium at SaITE 2016 (Webb et al., 2016): Matija Lokar, University of Ljubljana, Faculty of Mathematics and Physics, Slovenia, Peter Michaud, Alpen-Adria-University of Klagenfurt Institut für Informatik und deren Didaktik Universitätsstraße, Andrej Brodnik and Nataša Mori, University of Ljubljana, Faculty of Computer and Information Science, Slovenia.
4. All attendees of EDUsummIT 2015 for review and suggestions.
5. Attendees at the symposium at SaITE 2016 for comments and suggestions during and/or after the symposium.
6. Members of TC3 Committee for comments and suggestions during and after the TC3 meeting.

Current Membership of TC3 Task Force: "Curriculum - deeper understanding of roles of CS/Informatics"

Mary Webb (convener), King's College London, UK

Pieter Hogenbirk, Helen Parkhurst School, Netherlands

Nick Reynolds, University of Melbourne, Australia
Yaacov Katz, Michlala - Jerusalem Academic College and Bar-Ilan University, Israel
Andrej Brodnik, University of Ljubljana, Slovenia
Ivan Kalas, Comenius University in Bratislava, Slovakia
Maciej Syslo, UMK Toruń, University of Wrocław, Poland
Andrew Fluck, University of Tasmania, Australia
Torsten Brinda, University of Duisburg-Essen, Germany
Sindre Røsvik, Chair IFIP TC3, Norway
Niki Davis, University of Canterbury, Christchurch, New Zealand
Peter Micheuz, Alpen-Adria-University of Klagenfurt, Austria
Monique Grandbastien, University of Lorraine, Nancy, France
Johannes Magenheim, University of Paderborn, Germany
Sigrid Schubert, University of Siegen, Germany
Hans Frederik, VU University Amsterdam, Netherlands
Johan van Niekerk, Nelson Mandela Metropolitan University, South Africa.