WHAT IS THE PUBLIC FACE OF CODING IN AUSTRALIAN PRIMARY SCHOOLS?

John Hunt and Rose-Marie Thrupp
ICTs for Learning Consultant
Australia

Abstract

There is a perceived urgency to teach coding in schools from F-10. The premise most frequently used as the basis for this reasoning is future jobs, of which there are many proffered possibilities. It is therefore timely to consider where it (coding) fits, considering future workplace and career possibilities. The notion of coding being a *skill for all* has been present for almost two decades. Many have expressed a view that 20 years from now, maybe sooner, perhaps later, where post-school options are not yet known or determined, students will require a different skill set; on that there appears broad agreement. There is a wide commentary being proffered about what these futures are and how they are inextricably linked to digital literacies, and in particular coding.

The study sought to identify the extent to which schools made public their goals with regard to the place of coding in the curriculum. This was achieved by examining the websites of a sample of schools from each state and territory.

Introduction

Programming, now known as coding, has been taught in Australian high schools since the 1970s. Current documentation from various Australian education systems places a heavy weighting on coding from the early years of schooling. For example, *#Coding Counts* (State of Queensland, Department of Education and Training, Queensland, 2015) suggests that because the world of work is changing and that 3 in 4 of the fastest growing occupations require science, technology, mathematics and engineering (STEM) related skills and knowledge*,* there is a need to have "coding" explicit in the curriculum. This is a common response from other systems that have embraced coding in the curriculum. Though coding is represented in the Australian curriculum, Technology, the weighting is not necessarily consistent with that given other elements of the education system.

The essence of any educational movement is its implementation at a school level. As Australian primary schools grapple to respond to the challenge of teaching and learning of coding in the second decade of the twenty-first century, there is a need to review how the discipline of computer science is being valued and discussed in the school context. The interests and priorities of many outside of education, impact on the process of curriculum development for schools. In the case of coding, this has involved politicians, computer scientists, and industry with varied agenda. Less so has been the input from education.

Coding is and has been viewed under different guises. The history of teaching coding in Australian schools commenced in the 1970s; first limited to maths in high school and later, to the specific teaching of coding. Primary schools dabbled with Logo programming (Johnstone, 2003) in the 1980s, developed by Papert (1980) as a means of using the computer as an object with which to think. The current urgency to teaching coding in primary schools is different from these earlier days. It rests more with the reasoning that it is necessary for future jobs, of which there are many possibilities. It is therefore timely to consider where it (coding) fits, considering future workplace and career possibilities.

The notion of coding being a *skill for all* has been present for a considerable period of time. Many have expressed a view that 20 years from now, maybe sooner, perhaps later, where post-school options are not yet known or determined, students will require a different skill set; on that there appears broad agreement. There is wide commentary about what these futures are and how they are inextricably linked to Digital Literacies, particularly coding.

This review examines the many and varied conversations around coding in schools, primary schools in particular. It examines the work of commentators (social, media, political and industry), researchers, academics and policy makers. Whilst focused on understanding the Australian context, it also reviews the work of some international systems. Acknowledgement is made of the Australian state and territory responses where possible and endeavors have been made to examine individual school policy and practice across the respective states and territories of Australia.

The Literature About Coding

Programming, now called coding in education, draws upon the discipline of Computer Science. Jeanette Wing (2006) from the field of computer science defines computational thinking as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science.". She suggests that computer scientists then place these solutions within a computer language to be processed by a computer. Computational thinking describes the processes drawn upon when thinking about how a computer can be used to solve complex problems. This involves logical reasoning, algorithms, decomposition, abstraction, and patterns and generalization (Bers, Flannery, Kazakoof & Sullivan, 2014; Mannila et al., 2014).

These ideas from computer science have been translated into curriculum documents for education. From the perspective of the International Society for Technology Education (ISTE) (2016) a *computational thinker* develops and utilises strategies that show an understanding of how to solve problems using technology to develop and test solutions. In the ISTE Standards for Students, (Figure 1), a set of seven digital standards are proposed, of which computational thinking (coding) is one discrete element. Students are expected to develop a holistic understanding of each of these domains and how they might be used to resolve problems. This figure makes clear the connectedness of all domains without singling out computational thinking, connecting well with Wing's (2006) views that coding is a catalogue of thinking digitally.

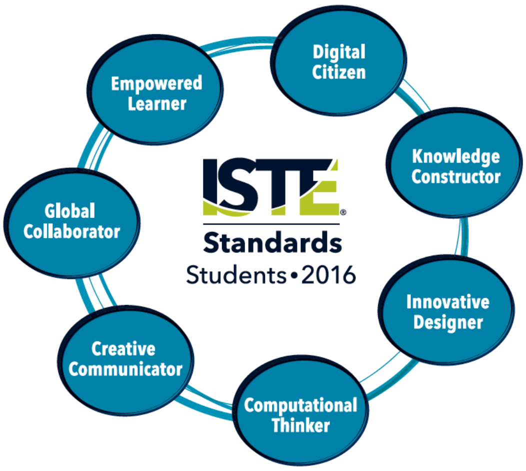

Figure 1. ISTE Standards for Students (ISTE, 2016).

In a glossary of the Australian Curriculum (n. d.), computational thinking is defined as:

> A problem-solving method that involves various techniques and strategies that can be implemented by digital systems. Techniques and strategies may include organising data logically, breaking down problems into parts, defining abstract concepts and designing and using algorithms, patterns and models.

Coding is viewed as one aspect of the Digital Technologies curriculum.

A Case for Coding in the Curriculum

In understanding the case argued for the inclusion of coding in the curriculum, it is necessary to consider the ideas of both commentators and researchers/policy makers. Commentators far outnumber the researchers/policy makers, and they are often linked to institutions and enterprises whose business is computer science and computer hardware/software.

The prospect of unknown future workforce needs has prompted a plethora of ideas about what work tomorrow might look like. We are at a crux in the 21st century where many are endeavoring to define what these jobs might be or what they might look like. They are inextricably linked to the rapidly developing and technological world we live in, but do they all require a sound knowledge of computational thinking?

The Commentators

Commentators bring various dimensions to the importance of learning to code. Journalists of online and paper documents enjoy the hype generated by writing about this topic. Often their contributions are generic and future work oriented. These vary from coding being a twenty-first century skill necessary for being a literate person (Crow, 2014; Prensky, 2008) to the professional opportunities provided by being able to code (Bradford, 2016). Crow suggests that it is a topic for all while others more specifically identify it as promoting the general skill of problem-solving. (Fractus Learning, 2017).

A recent report from the Foundation for Young Australians (FYA) (2017) *The New Work Smarts: Thriving in the New Work Order*, discusses areas of change where today's students need to be to cope with work in 2030. This report discusses the new skills required: Foundational skills, Technical skills and Enterprise skills.

The FYA (2017) clarifies these new work smarts and how they will impact on learning in Figure 2 (p. 4). Note the lack of specific reference to coding or computational skills.

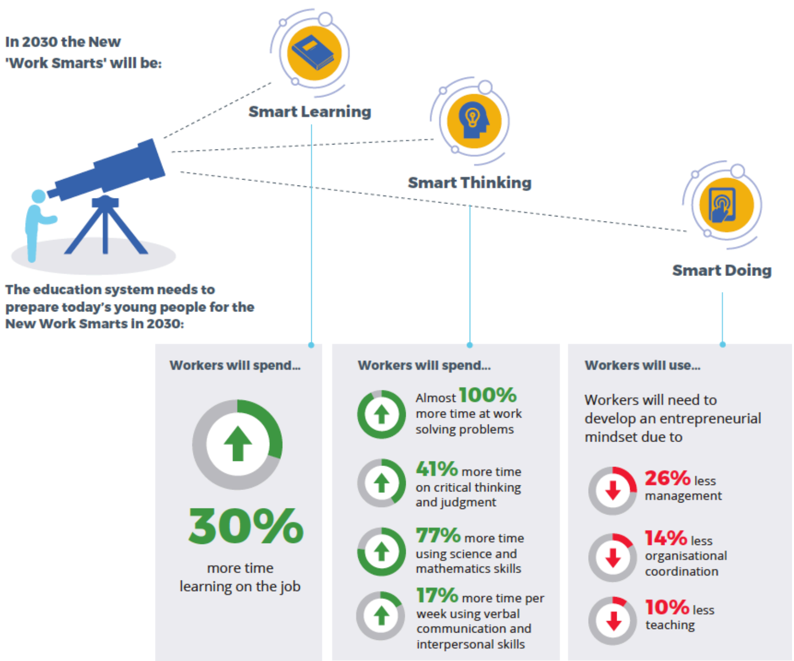

Figure 2: FYA Skills for the Future (Foundation for Young Australians, 2017)

In the list in Table1 drawn from *The Shape of Jobs to Come: Possible New Careers Emerging from Advances in Science and Technology (2010 – 2030)* (Talwar & Hancock, 2010), there is a shortlist of possible occupations of the future. Are there detailed references to coding or computational thinking in these descriptions?

Table 1

Future Occupations

1. Body Part Maker	10. Quarantine Enforcer
2. Nano-Medic	11. Weather Modification Police
3. Pharmer of Genetically Engineered Crops and Livestock	12. Virtual Lawyer
4. Old Age Wellness Manager / Consultant Specialists	13. Avatar Manager / Devotees - Virtual Teachers
5. Memory Augmentation Surgeon	14. Alternative Vehicle Developers
6. New Science' Ethicist	15. Narrowcasters
7. Space Pilots, Architects and Tour Guides	16. Waste Data Handler
8. Vertical Farmers	17. Virtual Clutter Organizer
9. Climate Change Reversal Specialist	18. Time Broker / Time Bank Trader
	19. Social 'Networking' Worker
	20. Personal Branders

The Researchers, Academics and Policy Makers

There is a worldwide momentum behind teaching coding with many countries experimenting with the inclusion of coding in the curriculum. (Sterling, 2016) This momentum has been generated by the Computer science and the Science, Technology, Engineering and Mathematics (STEM) sectors of the community. It varies in the reasoning: from work, global competition, and citizens being able to participate in daily life with some understanding of that which is controlling daily activities.

Common in this group is the fact that few will be coders in their jobs in the future but some aspect of coding knowledge will be required. Computer programming and coding will be a requirement in the digital economy, highlighting the global nature of the competition. Leon Sterling (Pro Vice-Chancellor: Digital Technologies, Swinburne University of Technology) identifies particular aspects of coding that will help students as critical thinkers able to solve problems, team players, designers of creative answers, innovators and entrepreneurs (in #Coding Counts, 2015. p.5) and strongly suggests coding to be the new literacy for all. Wilke (Senior Vice President, Cisco, Australia and New Zealand) suggests that innovators and entrepreneurs will need these skills to create new industries and new sources of wealth (#Coding Counts, 2015. p.4).

From a broader perspective, some policy makers argue for coding in schools from a *science, technology, engineering and mathematics (STEM)* perspective. (Committee for Economic Development of Australia (CEDA), 2015) This perspective cites coding as key to STEM learning and later working in STEM disciplines. Computer scientists (Crow, 2014; Wing, 2006) argue that computational think is an advantageous approach to thinking that can make contributions to many areas outside of Computer Science. These include representing problems, examining and critiquing multiple solutions to the same problem, error-identification and a readiness to respond to open-ended problems are generational skills of use across all STEM areas. (Bers, Flannery, Kazakoff & Sullivan, 2014) This aligns with the thinking that coding is about giving children the opportunity to engage with powerful ideas. (Papert, 1980) The computer just happens to be our era's best and most accessible tool for this purpose.

Some bodies such as International Society for Technology Education (ISTE) represent coding as having benefits well beyond the STEM agenda into other aspects of cognitive development. They claim the most important part of the K-5 coding experience is its ability to encourage and support creative expression and problem solving. Coding puzzles, tutorial progressions and unplugged activities (learning computing concepts without a computer) are all onramps to a world where students can be passionate and powerful enough to express their imaginations. Creativity, collaboration, persistence and abstraction are all thinking skills that coding contributes to.

Despite these reasons for learning coding in primary schools, a Google (2016) sponsored report, *Trends in the state of Computer science in the US*, suggested the growth of coding in schools developed minimally in the two years prior to the report. Principals, teachers and parents acknowledged the potential for coding programs in schools. Coding, however, is not evident in schools. Since 2015, the Australian National Curriculum has included a curriculum element, *Technologies,* of which Digital Technologies is one component. This component of digital technologies views coding as one aspect of learning in the digital ecosystem. Two years into this curriculum with politicians in various Australian states interested in coding's place in our schools, it is appropriate to study coding in Australian schools.

The Research Questions

This literature establishes that there is a wide interest from many sectors about coding in schools. However, the responsibility for teaching and learning related to coding rests with schools. Consequently, the context is evident to study the ways in which schools present their view of coding in learning for Australian children.

This study sought to answer these questions:

1. How is coding represented publicly by Australian schools?
2. To what extent do Australian schools place value on coding?

This provides a context for a qualitative study that subsequently examined one hundred and ten school websites.

Methodology

School websites are the location accessible to school communities and the public and therefore represent a public face of schooling. It can be reasonably assumed that websites are capable of providing evidence of that valued by the school community in relation to curriculum and learning. School websites are organised with a range of documents and statements.

This qualitative study used document analysis. The research interprets documents to give voice and meaning around a topic (Bowen, 2009). Document analysis, recognised as a social research method, is considered an efficient and effective research technique as there is generally a wide range of sources from which to draw. In this instance, websites provided a medium in which documents could be easily accessed: mission statements, strategic reviews, school improvement plans, newsletters, reports from national testing programs and other sources were reviewed. School websites were examined for reference to curriculum statements about ICT and Coding. This analysis was undertaken by the researchers. States and territories were represented proportionately, that is, selection was made as a proportion of schools in the Australian total. School ICSEA values (measure of community socio-educational advantage) were used to ensure a full range of schools from this perspective, urban/rural and school sizes were included. Both state and private schools were included. State and territory lists of all schools were compiled and by counting in fifties, schools were selected. A subsequent review of schools identified that all categories of schools were evident in the sampling. Further information about ICSEA values may be found at: http://docs.acara.edu.au/resources/About_icsea_2014.pdf

Websites were reviewed to locate references to coding in particular, Digital Technology as described in the Australian national curriculum and the use of Information Communication Technologies (ICT) as a pedagogical approach to learning. Investigating the websites required comprehensive searching of school annual reports, school improvement reports, vision and mission statements, photos, newsletters, events and lists of apps. This was not a linear process, requiring many null searches in websites. School websites are organized differently from state to state and from school system to school system. Search facilities on most school websites were untrustworthy. Manual searching was required, a lengthy process.

In examining school websites, consideration was given to the following data:

- Student enrolment numbers
- School ICSEA values
- Bring Your Own Device/Technology (BYOD/T) programs
- Explicit statements about Digital Technologies/ICTs for Learning
- Explicit statements about Coding

The researchers completed the study and analysis in November – December 2017.

Findings and Data

Table 2

School Data about BYOD, Technology Curriculum Statements and Coding

	Explicit Statement about Coding	BYOD/BYOT Program	Explicit Statement about ICTs for Learning
n=110	16 or 14.5%	17 or 15%	54 or 49%

Many school sites talked about digital resources with extreme differences noted in provision. Just half of schools investigated had statements (some inferred) about Digital Technologies and ICTs in learning. Only 15% of schools had a form of BYOD/BYOT program in place, most often limited to Year 4-6. Coding statements were found in just 14.5% of schools. Coding was most often only available as after school clubs or for a small select group: a broader coding curriculum was not evident.

Connectivity was evident in lists of ICT resources but not necessarily in the descriptions of actions or uses of Digital Technologies (DT). Teaching DT was not often a classroom learning priority. Teaching coding appears to have a similar low priority. Overall, Digital Technologies (and ICTs for Learning) has the appearance of a low public face.

Whilst a range of uses of DT and ICT was noted, it was unclear where it fitted into Learning and Teaching or the broader curriculum. It was most often noted in photos and references were evident in school reports or up front in Visions/Mission Statements. In the curriculum statements there was less evidence about how DT/ICTs for Learning were used.

If school websites were a public face, there would appear much that could be done to enhance their ability to communicate school priorities and a balanced curriculum, a curriculum that extends beyond Literacy and Numeracy. All schools had explicit Literacy and Numeracy statements.

It is clear that the use of ICTs in learning and teaching is developing in Australian schools with nearly half of the school websites including discussion on this topic, some intensively. The difference between these data and data about the teaching of Digital Technologies and coding may be a result of confusion by schools. There is a possibility that some schools and teachers view ICTs for learning and teaching as the same as teaching to the DT curriculum. Though the two can overlap, their intent is different as is the content to be taught. Generally, the confusion between the terminology DT, coding and ICTs for learning and teaching was evident in some schools and for some teachers.

The Research Questions Answered

1. *How is coding represented publicly by Australian primary schools?*

Despite the expectations of state and territory systems, the public face of coding is not very visible. Less than 15% of primary schools investigated had explicit references to coding. In these instances, references were often found in school newsletters, whilst it might have been expected that for a curriculum element given such prominence, it would have garnered a significant position in the unpacking of the school curriculum. However, noting that the broader curriculum was often not detailed at the school level (beyond Literacy and Numeracy), the researchers were not surprised at what they found. The curriculum section of a large number of schools websites examined showed little more than subject oriented booklists.

2. *To what extent do Australian schools place value on coding?*

One has to consider why coding has received such a high profile from systems, given its relative position in the Digital Technologies curriculum. It merits a few lines in the ACARA curriculum. There is an appearance that there is much hype around coding and the links to future workplace possibilities. It appears that commentators and commercial entities drive much of the coding agenda. Researchers on the other hand are more cautious, seeing that there are strong links to other agendas and ways of working, for example, links to STEM curricula and the cognitive challenges for all learners, not just those seeking a career that has not yet been invented. If schools are expected to invest so much time and funding in pursuit of a coding curriculum, they need to be absolutely clear as to why they are doing it. And this needs to be shared publicly. If a silo mentality is to be a part of schooling, the sharing of best practices and what works will continue to keep genuine innovation from emerging. It is the sharing of rich conversations between schools that enables substantial change to occur. The current low visibility of coding on school websites builds these silos and inhibits sharing.

Conclusions

The public-face of coding on school websites is sparse, that is not to say that coding is not being taught in isolated classrooms or in particular school programs. It could be said that the public-face of coding on websites is not consistent with the requirements of the current curriculum (ACARA and state/territory equivalents), and that the public-face of coding does not appear to align well with that aspired to by some government and policy documents.

Whilst systems have been quite explicit in detailing what a ‘digital curriculum’ might look like, it was not easy to glean much about this from school websites. This section of our document review has not borne the fruit expected, where it was expected rich stories about how schools are responding to the call to develop computational thinking or computational science (coding).

School websites appeared in the main to be formulaic, with generic templates supplied by systems and schools required to fill in the missing data in a localized context. Whilst this seems an ideal situation, in a great number of schools we found what might be considered important matters for a school's public face had either been removed, altered or retained with only a minimalist story available. Specific reference to the ACARA curriculum was not always evident and if present, scant detail of what each Key Learning Area (subject) looked like at either school or grade level was not evident.

The Way Forward

This study contains messages for systems about curriculum accountability When public funds are allocated to educational curriculum and related resources, there is an entitlement to know how the funds have been allocated, why they were allocated and the impact of this expenditure. This is of importance to those beyond a school community. There is also a need to inform parents, communities and others about how they (the schools) are responding to policy expectations. If coding is a genuine requirement of schooling, as policy suggests, then a higher visibility will be necessary.

Limitations of This Study

This study recognizes the problems caused by a small sample and the inability to access all school data. These limitations are however countered by the relative consistency of available data across the sites sampled and surveyed. The researchers maintain that the public face of schools is where parents and community would expect to find a comprehensive statement about a school and its curriculum.

References

Australian Curriculum, Assessment and Reporting Authority. (n. d.). Retrieved from https://www.acara.edu.au/curriculum/learning-areas-subjects/technologies.

Australian Curriculum. (n. d.). Retrieved from https://www.australiancurriculum.edu.au/f-10-curriculum/technologies//Glossary/?term=Computational+thinking

Bers, M., Flannery, L., Kazakoff, R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum in *Computers & Education, 72,* 145-157.

Bowen, G. A. (2009). Document analysis as a qualitative research method. *Qualitative Research Journal, 9*(2), 27-40. doi:10.3316/QRJ0902027

Bradford, L. (2016, June 20). Why every millennial should learn to code. *Forbes.* Retrieved from https://www.forbes.com/sites/laurencebradford/2016/06/20/why-every-millennial-should-learn-some-code/#ae333e070f24

Committee for Economic Development of Australia. (2015). *Australia's future workforce?* Melbourne, Australia: CEDA.

Crow, D, (2014). Why every child should learn to code. *The Guardian*. Retrieved from *https*://www.theguardian.com/profile/dan-crow;

Foundation for Young Australians (FYA). (2017). *The new work smarts: Surviving in the new work order*. Sydney, Australia: Author.

Fractus Learning (2017). *8 Reasons Why Kids Should Learn to Code.* Accessed from https://www.fractuslearning.com/2014/05/15/kids-should-learn-to-code/

Google Inc., & Gallup Inc. (2016). *Trends in the State of Computer Science in U.S. K-12 Schools*. Retrieved from http://goo.gl/j291E0.

International Society for Technology Education. (2016). Retrieved from https://www.iste.org/standards/for-students

Johnstone, B. (*2003). Never mind the laptops: Kids, computers and the transformation of learning.* Bloomington, IN: iUniverse Inc.

Mannila, L., Dagiene, V., Demo, B., Grhurina, N., Mirolo, C., Rolandsson, L., & Seattle, A. (2014, June). *Computational thinking in K-9 education*. Paper presented at ITiCSE-WGR, 14, Uppsala, Sweden.

Papert, S. (1980). *Mindstorms*. Sussex, United Kingdom: The Harvester Press.

Prensky, M. (2008). *Programming is the new literacy*: Retrieved from http://lablearning.eu/documents/doc_inspiration/prensky/programming_is_the_new_literacy.pdf .

Queensland Government. (2015). #*Coding counts: A discussion paper on coding and robotics in Queensland schools*. Retrieved from http://advancingeducation.qld.gov.au/SiteCollectionDocuments/Coding-and-robotics-booklet.pdf .

Sterling, L. (2016). Coding in the curriculum: Fad or foundational. *Research 2016 Conference Proceedings*. Melbourne Australia: Australian Council for Educational Research.

Talwar, R., & Hancock, T. (2010). *The Shape of Jobs to Come: Possible New Careers Emerging from Advances in Science and technology (2010 – 2030).* Fast Future Research. Retrieved from http://fastfuture.com/wp-content/uploads/2010/01/FastFuture_Shapeofjobstocome_FullReport1.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Author Details

John Hunt
johnhunt49@outlook.com.au

Rose-Marie Thrupp
rose-marie.thrupp@icloud.com