Changing Computer Curricula in Australia

Andrew E. Fluck[1(✉)] and Anit Girgla[2]

[1] Independent Educator, TAS Launceston, Australia
Andrew.Fluck@utas.edu.au
[2] University of Tasmania, TAS Launceston, Australia
Anit.Girgla@utas.edu.au

Abstract. This paper was developed from a symposium entitled 'Moving on with informatics/computer science curricula – challenges and opportunities.' It provides an Australian perspective for that debate. From 1901, the eight Australian States and Territories accepted constitutional authority for school education. Hence, a nationally developed curriculum only arose in 2014, for optional adoption. Computers were initially included as a general capability to enhance learning in all subjects. This focused on Office-suite applications. In 2016, the new Digital Technologies subject was added to the curriculum. This paper looks at the adoption of the new subject and changes proposed in the 2021 public consultation draft Review. We reveal the answers to the following questions about the proposed update for Digital Technologies: 1. Will Australia abandon the general ICT capability (as in the UK) to provide greater focus on Digital Technologies? 2. Will Australia put a greater emphasis on coding/programming than the current meagre mention? 3. What does a tutor of pre-service teachers think about the proposed changes?

Keywords: Digital technologies curriculum · Coding proportion · Pre-service teachers

1 Introduction and Context

1.1 A Nationally Developed Curriculum

Curriculum responsibility in schools was vested with the states of Australia by the constitution of 1901. A nationally developed curriculum for school subjects was only made available from 2014 and was incrementally adopted or appropriated by the states, in various ways.

At that time, computers were written into the nationally developed curriculum as a general capability to enhance learning in all subjects. This was designated as the Information and Communication Technologies (ICT) general capability. The core ICT capability was conceived as comprising Investigation, Communication and Creation. These elements are underpinned by 'managing and operating ICT' and 'applying social and ethical protocols and practices.'

Because different learning areas of the curriculum were developed along a staggered timeline, the Technologies learning area curriculum did not become available

Published by Springer Nature Switzerland AG 2022
D. Passey et al. (Eds.): OCCE 2021, IFIP AICT 642, pp. 144–155, 2022.
https://doi.org/10.1007/978-3-030-97986-7_12

until 2016. This contains the Digital Technologies subject. At its core is the concept of creating digital solutions, approached by processes and production skills. Underpinning these are Digital Systems and Representation of data.

Therefore, the Digital Technologies subject is quite separate from the ICT general capability. Very few teachers had computer education in their own schooling, or have encountered it in their pre-service training. While 'creating digital solutions' is core to the subject, coding or programming are mentioned very sparsely in the document. The design-time for this subject is 30 min per week in Years Foundation-2, 40 min for Years 3–4, an hour per week in Years 5–6, and 80 min per week in Years 7–8.

In 2020, the Australian Government agency responsible for the nationally developed curriculum announced the commencement of a Review. This review produced a public consultation draft of a new version of the curriculum in late April 2021.

The Review had the key task of simplifying the curriculum. Partly due to the staggered implementation, teachers initially focused on core subjects, and had felt overloaded as additional subjects such as Digital Technologies were released.

Recent indications are that principals and teachers confound the current ICT capability and the Digital Technologies subject. 30% of tools and websites used in Digital Technologies were seen to be content management systems, office suites and other generic tools rather than subject content specific software applications.

There are some key issues to be examined. First of all, there is tension between the two ways in which computers are used in Australian classrooms (general ICT capability and discrete Digital Technologies subject). It remains to be seen how these tensions will be addressed by the curriculum revision. Second, programming or coding is a key component in computing curricula in other countries – we will examine this aspect in the proposed revision of the Digital Technologies curriculum. Finally, it is useful to understand how teachers respond to the proposed revision. One of the authors had worked as a Digital Technologies Education tutor, and informally discussed the topic with some pre-service teachers. This article is now framed in such a way to find the answers to these three questions: 1. Will Australia abandon the general ICT capability (as per the UK) to provide greater focus on Digital Technologies? 2. Will Australia put a greater emphasis on coding/programming than before? 3. What does a tutor of pre-service teachers think about the proposed Digital Technologies curriculum changes?

After consideration of these research questions, we show how they illuminate the four focus points of the symposium on "Moving on with informatics/computer science curricula – challenges and opportunities". These were:

1. What is driving the emphasis of specifications for informatics/computer science curricula in different countries?
2. What do we know about how students learn some of the core concepts and processes of informatics/computer science that will enable us to design structure and progressions in curricula?
3. How should we incorporate new challenges associated with rapid developments in artificial intelligence and machine learning into informatics/computer science curricula?
4. What is the relationship between an informatics/computer science curriculum and other academic disciplines?

The responses to the three questions will shed light on the focus points of the symposium, as shown below.

1.2 The Proposals of the Review

The Public Consultation draft release in late April 2021 contained two main changes in the curriculum concerning student computer use:

1. The ICT general capability will be renamed as the 'Digital Literacy' learning continuum.
2. The Digital Technologies subject will grow from 43 to 71 Content Descriptors.

The Digital Literacy learning continuum [1] differs from its predecessor (ICT) in several ways. Digital safety and wellbeing replace social and ethical practices. Thus, there is a greater emphasis on cyber-safety and management of personal digital authentication tokens. Recognition of intellectual property, referencing and copyright is now balanced with exploration of creative commons, collaboration and information exchange. Figure 1 shows the proposed new structure and the new nomenclature.

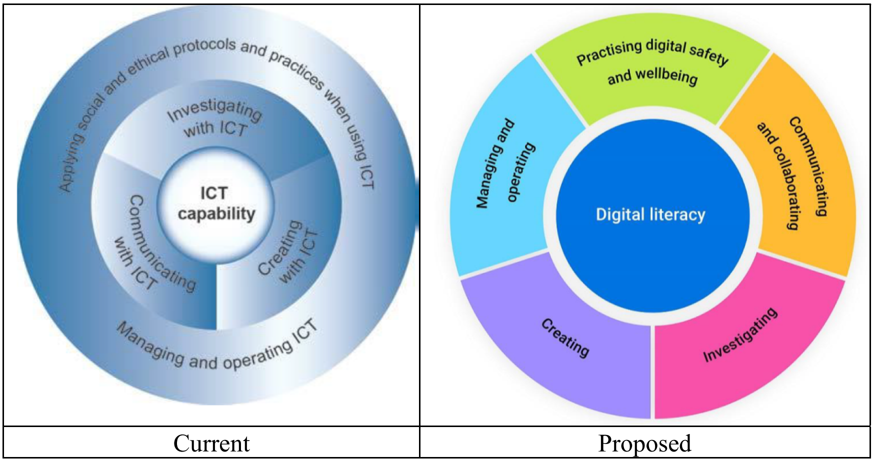

Fig. 1. The organising elements for the ICT general capability and the proposed Digital Literacy replacement (ACARA, 2016, 2021)

The Digital Technologies curriculum is due for a massive shakeup if the review proposal proceeds [2]. The current and proposed structures for this subject are shown in Fig. 2, with minor rearrangements of the elements.

The biggest change is a massive increase in Content Descriptors from 43 to 71. This appears designed to make them more understandable to teachers. More complex ideas have been broken into smaller components. For example, in Years 5–6 (student ages 10–12 years): "*Examine how whole numbers are used to represent all data in digital systems*" has been replaced by: "*Explain how digital systems represent all data*

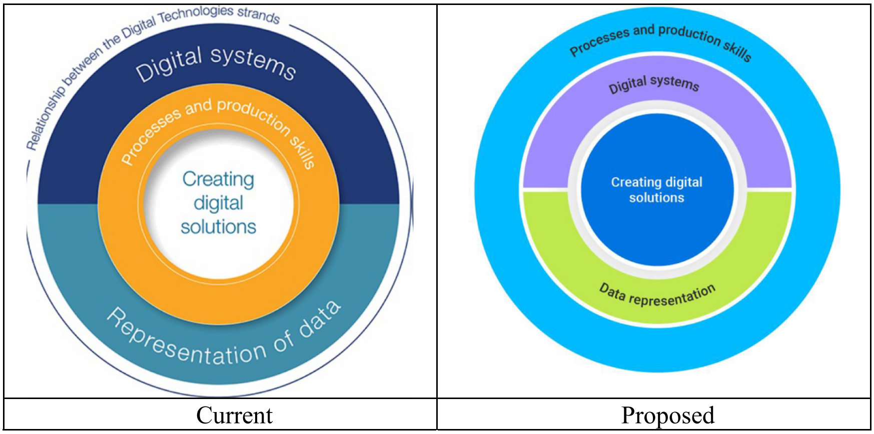

Fig. 2. The organising elements for the Digital Technologies subject (ACARA, 2017, 2021)

using numbers" and "*Explore how data can be represented by off and on states (zeros and ones in binary)"*.

This kind of simplification can help teachers by using simpler language while introducing otherwise hidden concepts (such as binary). The increase in more explicit language throughout the new proposal is likely to be well accepted by teachers.

Another noticeable change is the recognition that early childhood students (referred to as Foundation students, aged 5–6 years) have quite distinct learning needs from Years 1 & 2 (6–8 years old). These Foundation students are asked to engage with fewer concepts and use play-based learning. One content descriptor requires them to be taught to *identify some data that are personal and owned by them*. This is quite complex, but indicates a perception that cyber-safety awareness begins at a very young age.

2 The Issue of Coding

Having described the current status of computers in the Australian curriculum, and the impending changes, we now look at the issue of coding and compare this with the situation in other countries.

2.1 The Current Place of Coding

One of the consequences of the proposed increase in content descriptors for Digital Technologies is a proportionate reduction in coding-relevant teaching activities. The key concept in the subject is 'creating digital solutions'. The language used in the original curriculum does not ask students to code or program, but instead asks for them to *implement simple digital solutions* (Years 3–4).

In the overall Technologies learning area, computational thinking is fundamental to the subject. Computational thinking is defined as a "process of recognising aspects of computation in the world and being able to think logically, algorithmically, recursively and abstractly" [3]. The way digital solutions are to be created in the subject is through "using combinations of readily available hardware and software applications, and/or specific instructions provided through programming". It is quite clear from this language that coding or programming is just one of the methods students can employ.

To put programming/coding into context in the Australian Curriculum, it is useful to establish the format used. For each age range there is a description, a list of content descriptors (with elaborations), and the assessable achievement standards. Table 1 shows the current proportion of content descriptors in each age range where programming or coding is mentioned.

Table 1. Proportion of coding in the current Australian Digital Technologies subject

Student years	Total content descriptors	Content descriptors including programming or coding	% of coding in the subject
F - Year 2	6	0	0%
Years 3–4	7	1	14%
Years 5–6	9	2	22%
Years 7–8	10	1.5	15%
Years 9–10 (optional)	11	1.5	14%

Programming/coding is first mentioned with respect to visual programming in a content descriptor ("implement simple algorithms as visual programs involving control structures, variables and user input"), in the elaborations for Years 3 and 4 and in the achievement standards. Visual programming is a way of coding using coloured shapes on the screen. This recurs in Years 5 and 6.

In the current Australian curriculum, by Years 7 and 8, general-purpose programming/coding languages are mentioned in the description, and this occurs for the first time in content descriptors ("implement algorithms and modify and debug programs involving control structures and functions in a general-purpose programming language") at this level. "Use of a programming language" is also mentioned in the current Achievement Standards, so students are to be assessed on their programming skills for the first time once they reach this age (high school). In Years 9 and 10 (where Digital Technologies becomes an optional subject), we now see object-oriented programming/coding in the description, in two content descriptors ("implement, modify and debug modular programs, applying selected algorithms and data structures, including in an object-oriented programming language"), and in the achievement standards.

Overall, the current curriculum design has students progressing through the core F-8 Digital Technologies subject with visual coding in Years 3–6 and only undertaking general purpose programming/coding in Years 7 and 8.

2.2 Coding in Other Countries

In other countries, we see a different perspective on computer education in schools. In Singapore, for example, coding classes for primary school students were mandatory from the start of 2020 [4]. The initiative aims to develop an early appreciation of computational thinking and coding concepts—through simple visual programming.

While the number of content descriptors for the Digital Technologies subject in Australia is proposed to grow by 65%, coding/programming will decline from 14% to 8% of these. This proportional decline is in contrast to curricula in competing countries. For example, 27% of England's Computing attainment targets relate to coding, and 28% of progress outcome sentences in New Zealand involve programming.

In the New Zealand curriculum [5], coding is referred to as creating a 'Digital Outcome'. Reading through the supporting documentation to understand what this is, we find these kinds of explanations:

- Data management software like Filemaker Pro, Access or MySQL would be appropriate for this standard.
- Students iteratively develop, trial, and improve components in order to develop an increasingly refined outcome. During the development process, students are expected to describe and address relevant implications.
- The digital outcome must have been developed by the student.

It is not clear that creating a digital outcome necessarily implies writing code: however, it is also clear that doing so is one such method. The proportion of direct coding content in various national curricula can be seen in Table 2. It is clear there is no commonality in respect of the age at which coding should begin to be learned. There is also a range of coding proportions, even given the non-congruent age ranges.

Table 2. Proportions of direct coding in informatics/computing/digital technologies curricula

Country	Curriculum subject	Proportion of direct coding content
Australia (proposed)	Digital Technologies [F-8]	8%
Australia (current)	Digital Technologies [F-8]	14%
Singapore [6]	Computer Applications [Secondary 1–4]	17%
India [7]	ICT [Years 6–8]	18%
England [8]	Computing [Years 1–11]	27%
New Zealand	Technology – Digital Technologies [Years 1–13]	28%

For Australia, we believe there is an increased need for a foundation on computational thinking and coding at all levels, especially the upper junior, middle, and senior school curriculum. The current Digital Technologies subject shows a greater emphasis on teaching computational and programming/coding concepts in the higher junior school years. However, this decreases in the middle and senior school. We believe it should be similar across all ages of learning.

The Australian Digital Technologies curriculum also highlights, via the content descriptors for each stage, a significant mismatch. Teachers in the junior school who are not trained in programming, have more to deliver in this aspect compared to the specialist Digital Technologies teachers in the senior school, yet are required to be highly trained in this coding skill.

3 Views of Pre-service Teachers

Here are some considered opinions from a tutor of pre-service teachers who had nearly completed a unit on Digital Technologies when the public consultation draft for the reviewed curriculum was released. As previously mentioned, one of the authors was a Digital Technologies Education tutor. In the tutorial group, one pre-service teacher aimed to become a specialist computing teacher; half had a teacher parent or worked in a school; but none had attended a school placement because of Covid-19.

The tutor felt these pre-service teachers were ambivalent about ICT because priority had previously been given to the teaching of literacy, numeracy and science/inquiry. Although inquiry, deep and critical thinking could be promoted by teaching coding, Digital Technologies was not taught to them in school, and so was foreign to their thinking.

Through the informal discussions, the tutor formed three clear ideas about the proposed new curriculum for Digital Technologies. These concerned granularity – number & complexity of content descriptors; language – tension between clarity/specificity and abstraction/ambiguity; and year-on-year progression.

3.1 Granularity

When comparing the original curriculum with the proposed update, some content descriptors have been replaced by two. This change multiplies the total number of content descriptors, but generally splits complex learning activities into separate, more clearly defined activities. An example is shown in Table 3.

Table 3. Split content descriptor example, with distinction between pre-school and Years 1–2

Original	Recognise and explore digital systems (hardware and software components) for a purpose [Foundation to Year 2]
Proposed	Recognise and explore digital systems (hardware and software) and how they can be used to solve simple problems [Foundation/pre-school]
	Identify and explore digital systems and their components for a purpose [Years 1–2]

The original content descriptor has been segregated into two sub-components (see Table 4). While the split retained the essence of the original content descriptor, the distinction between Foundation and years 1 and 2 is an improvement because younger children need to have more specific content descriptors.

Table 4. Another split content descriptor – using more specific language

Original	Examine how whole numbers are used to represent all data in digital systems [Years 5–6]
Proposed	Explain how digital systems represent all data using numbers
	Explore how data can be represented by off and on states (zeros and ones in binary)

The proposed changes to the curriculum appeared to have reduced ambiguity, but not removed it entirely. Splitting an original content descriptor added clarity by breaking the topic into easy digestible parts. Table 5 shows use of specific language such as 'off and on – zeroes and ones in binary' made more sense than using the phrase 'whole numbers.' This leads onto a consideration of the language used in the proposed curriculum revision.

3.2 Language

In some cases, content descriptors in the revision have been re-written using different language (for example, shown in Table 5).

Table 5. A re-written content descriptor, to provide assistance for early career teachers

Original	Recognise different types of data and explore how the same data can be represented in different ways [Years 3–4]
Proposed	Recognise different types of data and explore how the same data can be represented differently depending upon the purpose

With a small change in the language, the revised content descriptor provides greater clarity. The proposed descriptor is more specific and would assist the teacher. Young teachers about to begin their careers would face a lot of uncertainties. If the curriculum was made more specific like this, they would have one less thing to worry about.

In other parts of the proposal, some new terms are introduced. 'User stories' occur in several content descriptors, but are not defined, which is a difficulty. In a couple more, students 'co-construct' products, which is a positive aspect of the proposal, leading to greater learner engagement.

3.3 Year-On-Year Progression

In some ways, the curriculum change had a mixed level of applicability. While some strands such as Collaborating and Managing had increased clarity for the lower years, the content descriptors had become less clear and briefer. Some sub-strands would not be applicable to science and maths teachers as they focused too much on programming, requiring dedicated lessons.

In respect of increased granularity, it was the tutor's opinion that some pre-service teachers perceived scope to build the digital curriculum around a spiral – giving children the advantage of linking back into foundational learning as they progressed towards deeper learning on the same topic.

To conclude, the tutor felt pre-service teachers who were going to teach lower grades felt better about the revised curriculum as it gave them more guidance. Pre-service teachers who were preparing to teach senior grades found the revised curriculum vague. While this is not a comprehensive approach to gauging the value of the revised curriculum, it nonetheless gives us a brief glimpse into the minds of pre-service teachers.

4 Discussion

4.1 Responding to the Research Questions

This discussion is framed around the three research questions posed at the start of this article. The first question addresses the tension between the general ICT capability and the discrete Digital Technologies subject. The proposed curriculum revision perpetuates both aspects but renames the former as Digital Literacy. There are some changes, such as the increased focus on cyber-safety in this general capability, but there is still some overlap in the areas of intellectual property and digital ethics. Unlike other countries which have abandoned the general IT skills aspects, Australia will continue to run both curriculum components. The renaming could accentuate pre-existing confusion between them, which could become a problem. Overall, there is hope that the new Digital Literacies capability and Digital Technologies subject will be more useful than their predecessors. However, the similar names are expected to exacerbate continuing confusion of these different aspects of student computer use.

Other countries have solved this issue by writing appropriate software tools into each subject curriculum and eliminating the ICT/Digital Literacy capability. This is beginning to happen in Australia, for example, dynamic geometric software is required in Mathematics at several Year levels. Another approach is to see that digital technology fundamentally changes the curriculum (for instance, there are no positive online translation skills embedded in the Languages subjects). However, it seems Australia is not yet at this point of curriculum transformation.

The second question looks at the place of programming or coding in the Australian Digital Technologies subject. Whereas other countries have over a quarter of their subject devoted to this topic, Australia proposes to diminish this from 14% to less than 10% of the subject. Curriculum designers appear to fear the explicit mention of programming/coding in the antipodes, masking this skill with terms such as 'create a digital solution' or 'digital outcome'. These very nebulous phrases may resonate with a teaching workforce largely untrained in coding, but also generate diverse alternatives which may not generate the meta-cognition implicit in programming. While the current authors shy away from multitudes of coding languages in schools, some diversity, clarity and logically conceptualised problem-solving culminating in coding can be seen as good preparation for active citizens.

Finally, our tutor discussions with pre-service teachers have highlighted tensions between granularity, language and progression. Teachers will feel comfortable in delivering good education if these issues can be resolved.

4.2 Responding to the Symposium Focus Points

This analysis from Australia sheds some useful light onto the four focus points of the symposium. The first focus point was on the impetus for informatics/computer science curricula specifications. As we have shown, the Review in Australia has led to greater granularity and clarity of language in the curriculum. Looking deeper, the reasons for actually having a Digital Technologies subject in the school curriculum is more complex. Day-to-day computer consumption/use by students was encapsulated by the ICT/Digital Literacies general capability at an early stage of the nationally developed curriculum. The more specialised and creative teaching of digital solutions resonates with economic rationales. While ICT/Digital Literacy contribute to greater gross domestic product, knowledge economies depend far more on digital innovation. The stated rationale for the Digital Technologies subject opens with: "*In a world that is increasingly digitised and automated, it is critical to the strength and sustainability of the economy, the environment and society that digital solutions are purposefully designed to include user empowerment, autonomy and accountability"*.

This rationale accords with OECD economic imperatives [9] to prepare citizens for participation in the knowledge society, and to become innovative creators for it. The rationale however goes further, incorporating environmental and social benefits. Finally, the rationale exhorts student personal achievement and responsibility.

The second question in the symposium enquires about student learning in this Informatics/Computer Science/Digital Technologies subject. Two observations can be made in the light of the Australian experience. At this stage, direct focus on programming is deprecated compared to curricula in other countries, with less than 8% of curriculum content descriptors related to coding. Instead, students use computer-based tools to create digital solutions. This is rather vague, but a lot of computer science skills and understandings are woven into the curriculum. The other observation that can be made is that coding instruction commences at age 9 (Years 3–4) with visual coding, and progresses to a general-purpose language at age 13 (Years 7–8). Object-oriented languages are optionally taught at age 15 (Years 9–10).

The third question asks about the incorporation of innovations such as artificial intelligence and machine learning into the curriculum. Many of these fall into the realm of non-deterministic computing. There are two issues to be considered: learning resources and teacher professional development. International companies have made some excellent learning resources for machine learning [10] and quantum computing freely available [11]. The Australian curriculum provides about 7 examples where students can engage with the use of artificial intelligence systems, but only one where they would work directly with such a system: "*…exploring artificial intelligence data* *analysis where an algorithm is trained by a structured dataset, for example, engaging with online machine learning examples (AC9TDI10P02_E2) [Year 10]".*

The final symposium question about the relationship between the Digital Technologies subject and the use of computers across the curriculum has already been answered. In brief, the relationship is one in tension, and fraught with misunderstandings by teachers, schools, students and parents.

5 Conclusion

The revised curriculum is due to be implemented from the start of 2022. Given the scope of changes to the use of computers in class, a great deal of communication and teacher training will be required.

The proposed revision to the Australian Digital Technologies subject appears to embed the rapidly changing nature of Information Technology. However, there is little mention of non-deterministic programming. Given recent government funding initiatives, artificial intelligence/machine learning and quantum computing might well be included in the Year 9–10 curriculum. The three take-home messages from this analysis are:

a) To what extent should the Informatics/Computing/Digital Technologies curriculum specification in each country be atomised? Greater detail can assist teachers with weaker content knowledge in the subject. Fewer specifications are inevitably phrased using more general language, which may facilitate innovation adoption.
b) The extent to which coding/programming is addressed in the subject is contested. Australia proposes to have a relatively small proportion of learning explicitly focused on coding (8%), which contrasts with the 25% or more in other countries. Perhaps an intermediate level is more appropriate?
c) Countries appear to differ in the framing of general ICT capabilities and dedicated Informatics/Computing/Digital Technologies subjects. Having both aspects in the Australian Curriculum seems to be confusing to teachers, but apparently will be a feature for the next five years.

As a final note of personal opinion, some pre-service teachers in our careers have submitted draft lesson plans for assessment, expecting school students to code with up to three languages in a single session. Understandably, these trainees are keen to demonstrate their capacity, knowledge and agility, but it does put a focus on how much coding is enough. The authors consider a maximum of four computer languages over the ages 5–16 should be sufficient, with only one taught in any year. This would provide practical experience to consolidate the other computational thinking skills acquired. Also, bearing in mind the Sapir-Whorf linguistic relativity hypothesis [12] that your language determines your thinking, these coding languages should have varied characteristics. Current curricula often portray computers as von Neuman uniprocessors or Turing machines programmed procedurally within an algorithmic paradigm. Machine learning, object-orientation and quantum scoring illustrate more parallel problem-solving processes which we believe should be part of each citizen's understanding of information technology.

References

1. ACARA (Australian Curriculum Assessment and Reporting Authority): Review of the Australian curriculum F–10: General capabilities - Information and Communication Technology (ICT) capability (Digital Literacy) (2021). www.australiancurriculum.edu.au/media/7024/gc_digital_literacy_ict_capability_consultation_curriculum.pdf
2. ACARA (Australian Curriculum Assessment and Reporting Authority): Technologies consultation curriculum – Digital Technologies – comparative information F-10 (2021). www.australiancurriculum.edu.au/media/7054/technologies_digital_technologies_comparative_information_f-10.pdf
3. ACARA (Australian Curriculum Assessment and Reporting Authority): The Australian Curriculum: Technologies, p.109 (2016). https://www.australiancurriculum.edu.au/download?view=f10. [select Digital Technologies > all year levels > all curriculum elements >Download PDF]
4. Tan, Z.: Singapore makes coding classes mandatory for primary school students, starting 2020. KrASIA (2019). https://kr-asia.com/singapore-makes-coding-classes-mandatory-for-primary-school-students-starting-2020
5. Ministry of Education (NZ): Technology in the New Zealand Curriculum (2019). https://nzcurriculum.tki.org.nz/content/download/168478/1244184/file/NZC-Technology%20in%20the%20New%20Zealand%20Curriculum-Insert%20Web.pdf
6. Ministry of Education (SG): Computer applications syllabus: Secondary One to Four (2020)
7. Central Institute of Educational Technology: Information and Communication Technology for the School System: Curricula for ICT in Education (2017)
8. Department for Education: Statutory guidance: National curriculum in England: computing programmes of study (2013). https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study
9. OECD (Organisation for Economic Co-operation and Development): ICT resources in school education: What do we know from OECD work? (2020). https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjSnsvX3JjyAhWG73MBHUCdAXQQFnoECAYQAw&url=http%3A%2F%2Fwww.oecd.org%2Fofficialdocuments%2Fpublicdisplaydocumentpdf%2F%3Fcote%3DEDU%2FEDPC%2FSR%2FRD(2020)2%26docLanguage%3DEn&usg=AOvVaw26MIEcWfGbQiS8qWburPu_
10. IBM & Dale Lane: Machine Learning for Kids (2017). https://machinelearningforkids.co.uk
11. IBM: Quantum Experience & Composer (2016). https://quantum-computing.ibm.com
12. Hojier, H. (ed.): Language in culture. In: Proceedings of a Conference on the Interrelations of Language and Other Aspects of Culture. Am. Anthropol. **56**(6) (1954). Part 2, Memoir No. 79