Teaching Computational Thinking in K-6: The CSER Digital Technologies MOOC

Katrina Falkner  Rebecca Vivian  Nickolas Falkner
School of Computer Science
The University of Adelaide
Adelaide, South Australia
Firstname.lastname@ adelaid e.edu.au

Abstract
In recent decades, ICT curriculum in K-10 has typically focussed on ICT as a tool, with the development of digital literacy being the key requirement. Areas such as computer science (CS) or computational thinking (CT) were typically isolated into senior secondary programs, with a focus on programming and algorithm development, when they were considered at all. New curricula introduced in England, and currently awaiting minister endorsement within Australia, have identified the need to educate for both digital literacy and CS, and the need to promote both from the commencement of schooling. This has presented significant challenges for teachers within this space, as they generally do not have the disciplinary knowledge to teach new computing curriculum and pedagogy in the early years is currently underdeveloped.

In this paper, we introduce the CSER Digital Technologies MOOC, assisting teachers in the development of the fundamental knowledge of CT and the Australian Digital Technologies curriculum component. We describe our course structure, and key mechanisms for building a learning community within a MOOC context. We identify key challenges that teachers have identified in mastering this new curriculum, highlighting areas of future research in the teaching and learning of CT in K-6.

Keywords: National curriculum, computer science, computational thinking, education, primary school, high school.

1 Introduction
Over the past decade ICT education has transitioned from focusing on ICT as a tool - with the development of digital literacy as the key requirement - toward understanding the underpinning concepts and workings of digital technologies. Areas such as Computer Science (CS) or computational thinking (CT) were typically isolated into senior secondary programs, with a focus on programming and algorithm development, when they were considered at all. The lack of computing curriculum at the primary level was perceived to be ‘failing to provide students with access to the key academic discipline of CS, despite the fact that it is intimately linked with current concerns regarding national competitiveness’ (Gal-Ezer and Stephenson, 2009).

To promote CS career pathways, global initiatives have targeted youth engagement and interest in CS through various outreach programs (Bell et al, 2011; Koppi et al., 2013; Lambert & Guiffre, 2009; Liu et al, 2011; Myketiak et al 2012). However, research findings and a continued lack of uptake of CS degrees suggest that outreach programs have had little success (Koppi et al., 2013). More recently, a drive to include computing in schooling curriculum has arisen, proposing that all children should have an opportunity to develop CT skills and have opportunities to be ‘creators’ of digital technologies (Gander et al., 2013; The Royal Society, 2012).

New curricula introduced in England (Department for Education, 2013), Australia (ACARA, 2012), New Zealand and the new ACM CS standards (Seehorn et al., 2011) have identified the need to educate for both digital literacy and CS, and the need to promote both learning areas from the commencement of schooling through to high school, to support the future generation of digital creators and increase international competitiveness. This is a significant milestone yet also raises a number of challenges, including the preparation of teachers and development of resources to support the success of implementation at a national scale. Curriculum change is not easy for teachers, in any context, and to ensure teachers are supported, scaled solutions are required. A potentially key factor in the success of implementing a new computing learning area will be appropriate professional development (PD) that provides teachers with the confidence and experience to integrate CS effectively into their classroom activities.

One educational approach that has gained traction for delivering content to large-scale audiences are massively open online courses (MOOCs), however, little is known about what constitutes effective MOOC design; particularly within the contexts of CT and teacher professional development. In this paper, we introduce the CSER Digital Technologies MOOC, assisting teachers in...
development fundamental knowledge of CT and the Australian Digital Technologies curriculum component. We describe our course structure, approaches to teaching CT within the K-6 context, and key mechanisms for building a learning community within a MOOC context. We identify key challenges that teachers have identified in mastering this new curriculum, highlighting areas of future research in the teaching and learning of CT.

2 The Australian National Curriculum

The Australian primary and secondary curriculum system is undergoing a significant period of change, with the introduction of a National Curriculum. In Australia primary school includes the first year of school, called Foundation (F), also known as Kindergarten (K), until year 6 or 7, depending on the state. Secondary school (also known as high school) includes years 7 or 8 to year 12. The Australian Curriculum describes the nature of learners and curriculum across three broad year-groupings: Foundation to Year 2 (ages 5-7); Years 3 to 6 (ages 8-11); and Years 7 to 10 (ages 12-16).

In 2013, the Australian Curriculum Assessment and Reporting Authority (ACARA) released a series of draft curriculum standards for the national curriculum to be introduced across Australia in 2014. The curriculum introduces new learning areas with considerable effort committed in the definition of the curriculum and national achievement standards for each area. Some learning areas have achievement standards defined from K-12, while others, including ICT, have achievement standards defined from K-10, with decisions in the senior years of schooling to be defined at a later stage.

‘The Shape of the Australian Curriculum’ (ACARA, 2012), identifies that ‘rapid and continuing advances in ICT are changing the ways people share, use, develop and process information and technology, and young people need to be highly skilled in ICT’. The ACARA documents include ICT awareness (digital literacy) as a key capability, embedded throughout the curriculum, and introduce a new learning area, Technologies, combining the ‘distinct but related’ areas of Design and Technologies and Digital Technologies (DT) (ACARA, 2013). DT focuses on developing knowledge of digital systems, information management and the CT required to create digital solutions. The core is the development of CT skills: problem solving strategies and techniques that assist in the design and use of algorithms and models.

The DT curriculum does involve some (CS) knowledge and skills, as well as some digital solutions (possibly involving programming and CS concepts) but the intended focus is on developing computational thinking, logic and problem solving capabilities. The DT curriculum is based on a systems thinking approach, designed to encourage students to understand the individual parts of the system, while also being capable of having a holistic view of the, including ethical, societal and sustainability considerations.

DT focuses on developing knowledge of digital systems, information management and the computational thinking required to create digital solutions. The core is the development of computational thinking skills: problem solving strategies and techniques that assist in the design and use of algorithms and models. The Australian Curriculum describes the nature of learners and curriculum across three broad year-groupings: Foundation to Year 2 (ages 5-7); Years 3 to 6 (ages 8-11); and Years 7 to 10 (ages 12-16).

Approaches to teaching vary according to the curriculum year-groupings. The development of both digital literacy and CT commences in the K-2 band and learning is based around directed play, facilitating students in developing an understanding of the relationship between the real and virtual worlds, the use and purpose of technology in communication, and the importance of precise instructions and simple problem solving in the digital world. In Years 3-6, students are guided to develop a wider understanding of the impact of technology, including family and community considerations, and are able to work on, and communicate about, more complex and elaborate problems. In this year level, students begin to apply CT to develop algorithms with visual programming software. Across Years 7-10, students move beyond their initial community and are required to consider broader ethical and societal considerations. In this band, students should be able to solve sophisticated problems using technology, and understand complex and abstract processes. Students begin to apply CT in their use of general-purpose programming languages to solve problems and create digital solutions. This development from K-10 supports the understanding of the utility of technology, as well as the development of problem solving skills and an abstract understanding of CS.

The eight key concepts that underpin the DT curriculum are allocated to one of two strands: ‘Knowledge and Understanding’ and ‘Processes and Production Skills’.

2.1.1 Knowledge and Understanding

The Knowledge and Understanding strand builds awareness of digital systems and digital information. This includes the impact of digital technologies upon societies and relationships between these technologies and a society, exploring ethical and cultural considerations, from both a local and global perspective. The following sequence of learning objectives explores how an understanding of digital representation is developed across the curriculum:

- K-2: Recognise and play with patterns in data and represent data as pictures, symbols and diagrams.
- 3-6: Explain how digital systems represent whole numbers as a basis for representing all types of data.
- 7-10: Explain how text, audio, image and video data are stored in binary with compression.

2.1.2 Processes and Production Skills

In Processes and Production Skills, students explore how to solve computational problems, involving developing skills in ‘formulating and investigating problems; analysing and creating digital solutions; representing and evaluating solutions; and utilising skills of creativity, innovation and enterprise for sustainable patterns of living’ (ACARA, 2013).
The following presents an example sequence of learning objectives designed to introduce algorithmic planning:

- **K-2:** Follow, describe, represent and play with a sequence of steps and decisions needed to solve simple problems.
- **3-4:** Design and implement simple visual programs with user input and branching.
- **5-6:** Follow, modify and describe simple algorithms, involving sequence of steps, decisions and repetitions that are represented diagrammatically and in plain English.
- **7-8:** Develop and modify programs with user interfaces involving branching, repetition or iteration and subprograms in a general-purpose programming language.
- **9-10:** Collaboratively develop modular digital solutions, applying appropriate algorithms and data structures using visual, object-oriented and/or scripting tools and environments.

The processes and production strand encapsulates the key concepts of CT and presents challenges to us as a community in how we develop relevant skills within the younger age groups.

### 2.2 Challenges of New Computing Curriculum

The challenges faced by both nations in the adoption of these curricula are extensive. Consultation with Industry, Community and Education within Australia (ACARA, 2013b) has identified significant concerns in relation to teacher development (particularly at K-7), appropriate pedagogy, and skills needed for integration of DT learning objectives with the teaching of other learning areas. Respondents (55%) indicated concern with the manageability of the implementation of the DT curriculum and 45% of respondents did not think that the learning objectives were realistic. Further concerns were expressed regarding teacher preparation.

Bell et al (2012) describe the New Zealand experience of the rapid introduction of a senior secondary CS curriculum, and the need for extensive teacher development that addresses both content knowledge and pedagogical knowledge. In the Australian curriculum, this will involve teachers understanding CT, CS disciplinary knowledge as well as the development of skills in visual or general-purpose programming. Further, it has been recommended that key to teacher development will be the integration of aspects such as CT across other learning areas (Yadav et al, 2011). In their CT course for educators, the instructors recommend incorporating CT modules into teacher education courses to expose teachers to these ideas. Through connecting CT to learning areas, it is recommended that teachers will be able to move beyond an ‘abstract’ idea of CT and understand its application and relevance as a problem-solving tool. Ragonis, Hazzan, and Gal-Ezer (2010) identify best practice as the development of a dedicated teacher development programme specifically addressing CS. They recommend that a critical element of such programs is to use empirical research to guide appropriate pedagogy for specific year bands, and learning objectives.

However, despite materials being available to teachers through PD, Settle et al. (2012), recognise the difficulty teachers face in translating materials into existing curriculum, when unfamiliar with the tools. In a study by Meerbaum-Salant et al (2011), they identified that even teachers experienced in CS, can be challenges with the introduction of new tools, which created feelings of anxiety, and resulted in teachers to deviate from original lesson plans. Another issue regarding tools is that they may be suggested to teachers to use to teach subject matter but they may not always be available. Tinapple, Sadauskas, and Olson (2013) further comment on the challenge for teachers, where expected software and/or hardware are not easily available. This is a consideration that needs to be taken into account with national computing PD, particularly when teachers from a variety of contexts (e.g. rural, disadvantaged) may be participating. Such findings indicate that teachers require opportunities to explore tools and also alternative ‘unplugged’ lessons as well as a variety of potential software that could be used.

In our previous review of research in the teaching and learning of CT within K-12 (Falkner, Vivian & Falkner, 2014), we identified a dearth of research into the development of appropriate pedagogy within the K-10 space, and in particular, within the K-6 space, with most of the research that has been done is situated within outreach programs, focussed on sharing teaching techniques aimed at motivating students to study CS, to address negative perceptions of the discipline, stereotypes and to increase diversity in our student cohorts. This places extreme pressure on deliverers of PD as well as teachers, when pedagogy and pedagogical strategies are underdeveloped in the K-10 space of computing education.

Developing pedagogically appropriate lessons for particular contexts, needs and students may be challenging for teachers and the adoption of teaching approaches may be influenced by teacher confidence in teaching the learning area. In one study, when teachers used guiding activity resources for their CS lessons, they were apprehensive about using teaching methods such as group work (Curzon et al, 2009). Further, teachers felt that because they were unfamiliar with the topic, considerable preparation would be required. In Black et al’s survey (2013), they discovered that teachers tend to focus more on fun activities rather than providing opportunities for deep learning of CT, focussing on impressive technology, physical computing and programming in constructionist environments. These forms of activities can complicate the learning environment further by placing additional stress on teachers inexperienced with technology.

Support for the professional development of teachers is crucial in expanding CS curricula, including the creation of community networks to share insights and pedagogical approaches and research (ACARA, 2013; Gander et al, 2012). This was confirmed by a study by Black et al. (2013) involving a survey of UK computing teachers in relation to their suggestions on improving CS education, and teacher development needs. Although their results highlighted a need for teacher training, they also expressed the need for a network and community to support resource development.
3 Massive Open Online Courses

Massively open online courses (MOOCs) offer one means to deliver education at scale and have the potential for community elements to connect participants across various locations, even around the world. Although online learning is not new, it has been argued that the difference between online learning and MOOC environments are the combination of teaching approaches course instructors use, the massive levels of participation and the openness (Glance, Forsey, & Riley, 2013).

Typically two different types of MOOCs have been identified; one being based on courses that embrace the use of videos to deliver content and computer-assisted online assessment (‘xMOOCs’) (Glance et al., 2013) and other courses based around online communities and connectivist principles, called ‘cMOOCs’ (Siemens, 2005, 2012). A number of ‘hybrid’ MOOC versions are also surfacing, that combine a mixture of xMOOC and cMOOC approaches, blending a structured pace with a focus on participant-led communities, such as EDMOOC by Coursera, and MOOC-EDs introduced by the Friday Institute (Kleiman, Wolf, & Frye, 2013).

Enrolment in MOOCs have reported significantly high enrolment rates, with edX and MITx reporting a total of 841,687 registrations from the fall of 2012 to the summer of 2013 across a number of their courses (Ho et al., 2014). In that year, 43,196 participants earned completion certificates. On average there was a 5.17% completion rate across the courses, with a 9% completion rate for those who went beyond ‘enrolment’ in the course. A typical measure of completion within xMOOCs is the completion rate for those that complete half or more of the course, known as explorers – edX and MITx report a completion rate for explorers of 54%. A supporting component of xMOOCs are the community forums, which have seen engagement anywhere from 6.5% to 25.7% with an average of 7.9%.

In comparison, cMOOCs measure enrolment based on members who ‘subscribe’ to the course via mailing lists or by signing up to the course platform. cMOOC enrolment figures have been found to be ranging from the hundreds to the low- thousands and researchers typically report participant engagement through the measurement of social media activity (de Waard et al., 2011). While the communities engagement seems large and broad, analysis of cMOOC social media engagement reveals that typically a small core of participants generate the activity. For example, in CCK11, 18% (N=126) participants were actively involved (Kop, Fournier, & Mak, 2011) and in First Steps in Teaching and Learning (FSTL12) (Roberts, 2012) about 30% actively participated throughout the 6 weeks and only 14 participants undertook assessment and received a certificate.

4 The CSER Digital Technologies MOOC

In selecting a ‘hybrid’ MOOC approach, we were able to deliver structured content as well as adopt a participant-led community, which is proposed as being valuable for teacher support in computing curriculum implementation (ACARA, 2013; Gander et al, 2012; Black et al. 2013). A large focus of the Australian DT curriculum is on CT, which is defined in the ACARA curriculum documents, as ‘a problem-solving method that involves various techniques and strategies, such as organising data logically, breaking down problems into components, and the design and use of algorithms, patterns and models’. Understanding CT involves understanding core CS concepts, and the ability to conceptualise and create abstractions that define solutions to problems.

At the level of K-6, the teaching and learning of CT involves the developing of capabilities in solving problems, utilising core concepts such as algorithm definition – including the introduction of selection and iteration – and data collection and analysis. Also introduced are key ideas such as abstraction and decomposition. Previous work in educator PD recommends integrating new concepts throughout courses and the application of concepts to other learning areas (Yadav et al. 2011). CT concepts and ideas were presented throughout the MOOC modules and examples of the concepts (e.g. abstraction, decomposition) were defined, incorporating lesson ideas with application to everyday examples and other learning areas.

4.1 Course context

The average age of primary teachers is 42.1 and 44.5 for high school teachers, with leadership roles being held by those around 50 years of age (Cordova, Eaton, & Taylor, 2011). In Australia, the teacher workforce is predominately female, particularly in the primary years (81% of primary teachers and 57% for secondary teachers). In Australia, teachers are reportedly spending 46 hours per week on all school related activities and about 8 or 9 days a year toward professional learning (Cordova et al., 2011).

Australian primary school teachers are typically generalist teachers, with 80% reportedly teaching in generalist classrooms (Cordova et al., 2011), implementing the various learning areas prescribed by their state or territory. Some schools are fortunate enough to have specialist teachers, such as an ICT teacher, but this is not typically the case for all schools, with only 6% (N=7,500) of teachers reportedly teaching computing (Cordova et al., 2011). In Australia, 17% of teachers report having had some post-secondary education in computing, with only 8% having been trained in the practice and pedagogy of computing (Cordova et al., 2011). Teachers are typically left to integrate the use of ICTs and digital literacy into their classroom activities by integrating with other learning areas.

4.2 Course Structure and Design

In response to existing research findings, we identified in the development of this course the importance of providing learning and teaching opportunities that were tool-independent and focussed on deep learning (Black et al, 2013, Meerbam-Salant et al, 2011), and the need to provide exemplars of activities that were already integrated with existing knowledge areas within the curriculum – removing the need for direct translation (Settle et al, 2012). We drew on and adapted existing lesson ideas from organisations and initiatives such as CS Unplugged and Code.org, and drew on lesson ideas and approaches from education texts in other learning areas, such as Mathematics, Science and Literacy, and with
examples from possible teaching themes, commonly used within K-6.

As a new learning area, the disciplinary content would be new for many teachers. Therefore, the course was designed around a series of seven topics that align with the Australian curriculum, delivered in a logical order, suitable for someone learning CS for the first time over a period. Our goal in the first unit was to provide an introduction to digital technologies, showcasing the development and application of digital solutions to solve real-world problems. Further, we wanted to define terminology for digital technologies (e.g. computing and CS) and distinguish between digital literacy and digital technology creation and CT. In unit 2, the more familiar topics of patterns (creating and continuing sequences and recognition) and data representation (collecting and representing data in different ways, with and without technology) were introduced because of the potential links to what teachers are already doing in Mathematics and Science. In subsequent units, we moved toward the use and application of data by computers and digital data as well as the introduction of more abstract concepts, such as algorithms. Lastly, we visited visual programming environments.

In each unit, the topic (e.g. ‘digital systems’) was introduced with the relevant Australian learning objectives. Each unit were broken into sub-topics and for each sub-topic a concept video was created or an existing suitable video used in which the concept was explained and supported with analogies and real-world examples. Links were made to the Australian curriculum ‘expected outcomes’ as guiding points for assessment. The goal of the course was to deliver core disciplinary knowledge, packaged for primary year levels, and lesson ideas so that teachers could feel comfortable and empowered to create or draw on existing resources to design learning activities to meet the learning objectives. The sequence of units for the Digital Technologies course are outlined in Table 1.

Table 1: Sequence of MOOC Modules

<table>
<thead>
<tr>
<th>Unit 1: Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 2: Data – Patterns &amp; Play</td>
</tr>
<tr>
<td>Unit 3: Data - Representation</td>
</tr>
<tr>
<td>Unit 4: Digital Systems</td>
</tr>
<tr>
<td>Unit 5: Information Systems</td>
</tr>
<tr>
<td>Unit 6: Algorithms &amp; Programming</td>
</tr>
<tr>
<td>Unit 7: Visual Programming</td>
</tr>
</tbody>
</table>

The core elements of the course were the focus on worked examples, and the sharing of further examples as identified and developed by the teachers themselves. To assist in community development, we initiated a Google+ community for the course; this space allowed participants to network, share ideas and course tasks and to collectively build an online series of resources corresponding to topic areas.

Each unit incorporated two fully worked examples of how specific learning objectives could be addressed across K-6. For example, Unit 6: Algorithms & Programming, incorporated a worked example exploring instructions and sequences of instructions addressing the learning objectives (with ACARA id):

- Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems (ACTDIP004)
- Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010).

Within this worked example, we explore multiple learning and teaching activities for different age groups connected to different knowledge areas, starting with a wriggle break activity, commonly included to signpost a change between activities in early years. A variety of instructions are written on pop sticks, such as “jump up and down”, or “spin around”. A student or the teacher selects a pop stick at random, and the class acts out the selected activity, with opportunities for paired or group exploration to demonstrate achievement. This activity is deceptively simple – while fun, and engaging for the students, it introduces the idea of instructions, and sequences of instructions through variants of the game. We then explore an extension of this activity, designed to assist in literacy development, where students construct a sentence, word by word, through rearranging a series of pop sticks (see Figure 1).

Figure 1: Computational Thinking development embedded within Literacy (Tunstall, 2013).

We continue this worked example, by exploring learning and teaching activities where students are able to construct their own instructions, incorporating existing videos and examples, including the well known “jam sandwich” example (Bagge 2012), and identify ideas for assessment of these learning objectives:

- Students understand that computers require explicit instructions.
- Students can explain that an algorithm is a step-by-step sequence of instructions.
- Students can re-order instructions or develop instructions that form a logical sequence.
- Students can adapt instructions based on their observation of an outcome.
- Students use descriptive and precise language when giving instructions.
- Students can provide a set of instructions to achieve a desired outcome.
For each unit, teachers were asked to post a task on the Google+ community page for the course. These tasks were designed to be informal and promote the exchange of tools, resources and lesson ideas. In all cases, teachers were provided with three options so that we could have a variation of content being shared. For example, within the same unit, we considered: ‘Design an activity that explores sequences of instructions’.

As an indication of the type of engagement within the course, we include here two example activities shared by teachers within the course. Teacher A integrated a lesson within the context of Mathematics, incorporating Beebots as an interactive technology at Year 1; the activity required students to identify the sequence of instructions to navigate their Beebot between two points on a map. This activity required the students to develop and demonstrate key skills in problem solving, instruction selection, sequencing and navigational language.

Teacher B identified a lesson activity that could be either represented on its own, or within the Drama learning area in the Arts curriculum at a more senior year level (Years 3 or 4), where students were asked to define a sequence of instructions for a random scenario, such as brushing teeth, or packing a school bag, which were then enacted by the group in a performance. This activity required the students to develop and demonstrate key skills in problem solving, instruction design and selection, sequencing, selection and repetition. Further integrating aspects of digital systems, they suggested a simplification of the activity suitable for Year 1, adopting QR codes as a mechanism for accessing a set of existing instructions for a given scenario.

Teachers were able to comment on peers’ task contributions in the Google+ community by providing constructive advice or suggesting extensions or ways that the activity could be adapted for other learning contexts.

Of the 1374 people who enrolled in the course via the course website, we had 473 participants connect to the Google+ community. Counting Google+ community posts of the community revealed that posting activities began at 269 for unit 1 and tapered off to around 100 in for the portfolio.

![Number of Google+ Community Posts](image)

While we felt a level of enthusiasm from the community participants appeared to be actively posting, what was participation and engagement like across the course and how did teachers experience learning digital technologies content and partaking in the course? The follow section reports on data obtained about participant engagement and survey findings about participant experiences.

### 5 Cohort Participation

Although we did not originally ask participants for demographic details, we were able to gather some idea of where participants were located via an anonymous survey as part of an optional activity in unit 3. The majority of participants appear to be from South Australia, Queensland, New South Wales and Victoria (see Figure 3, N=174). Advertising and visits generally covered these areas, suggesting that for future courses, more targeted advertisements and connections need to be made to Western Australia, Northern Territory and Tasmania.

![Survey results for location of participants](image)

Unsurprisingly, with the majority of teachers being female in Australia (81%) and the average age of teachers being between 40 and 50, according to the YouTube analytics, the majority of the cohort was female and between the age bracket of 45 to 64 (see Figure 4). These results show that we were able to target our intended audience and attract a female demographic that is typically lacking in post-secondary courses and careers (Koppi et al., 2013).

![Google Analytics demographic details for video views](image)

Of the 1378 enrolled in the course, 99 participants completed the course and 438 did not engage in the course any further than enrolling. As a result, we have a 7.2% completion rate, or 10.5% completion rate for those who went ahead and began the course. When considering completion rates, and measures of MOOC engagement, we consider engagement across all course components, and within core components specifically. Our completion rate overall was 7.2%, with a further 5.73% of participants exploring half or more of the course (without completion), and 56.39% of the participants completing less than half of the course. In terms of core components only, 8.13% of the participants explored half or more of the core components (without completion) and 52.3% of
the cohort (group of MOOC participants) explored less than 50% of the core components. Our completion rate for explorers was 55.7%, and 46.9% when considering core components only.

Overall, across the course platform and the Google+ community, the completion rates were mostly in-line with what one would expect to see in MOOCs in terms of enrolment and completion. However, 34.3% of the cohort (n=473) viewed and/or engaged with the online community – a significant increase in engagement over typical MOOCs. The completion rate relative to those that engaged with the community is 20.9%. A key motivating factor for this engagement was tying the course tasks in with the use of the Google+ community – a strategy that resulted in the co-creation of K-6 digital technologies resources and lesson plans.

In accordance with the participation and engagement described previously, we had a high number of viewers watching videos during the first unit (N=462), slowly decreasing during each module (to 66 in unit 7). According to the YouTube analytics, the average video length created by the CS Education Research group, was 5.8 minutes – ranging from around 1 minute to 11 minutes. This timeframe is typical of many of the xMOOC style video length. The average length of the videos watched was 4.37 minutes, suggesting that short concept videos work; however, designers need to consider presenting important information at the beginning.

5.1 Survey Responses

5.1.1 Survey: Participant Background

Some 51 participants responded to a survey about their experience in the MOOC. 45 females and 5 males responded. We recognize that the sample is biased towards educators who completed the course but the participants are able to provide insight into their experience across the whole course. Almost 50% (23%) of survey participants said that they had not participated in an online course before (28 had). A majority of the survey participants were teachers in primary schools, with some people in leadership roles or enrolled in university pre-services courses (see Table 2).

We received survey responses from most people throughout Australia, except from the Northern Territory and Western Australia. Those who did respond were from Queensland (16), Victoria (11), New South Wales (10), South Australia (7), Tasmania (4), Australian Capital Territory (2) and one from the United Kingdom.

The MOOC was targeted at the Foundation to Year 6 levels, however, we had a number of people enroll who were not specifically focused on these years in their professional roles. Table 4 shows that survey participants were primarily from the K-6 years but also worked across multiple year levels or upper year levels.

Table 3: Survey Participants’ Year Level

<table>
<thead>
<tr>
<th>Year Levels</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-6 Years</td>
<td>19</td>
</tr>
<tr>
<td>Years 7-12</td>
<td>8</td>
</tr>
<tr>
<td>Multiple</td>
<td>16</td>
</tr>
<tr>
<td>Higher Ed/ Other</td>
<td>2</td>
</tr>
<tr>
<td>Not a teacher</td>
<td>3</td>
</tr>
<tr>
<td>Missing</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>51</td>
</tr>
</tbody>
</table>

Table 3 demonstrates that survey respondents were from a variety of different Year levels, with most from either the K-6 years or working across multiple years.

Table 4: Survey participants’ confidence before the MOOC

<table>
<thead>
<tr>
<th>Experience Teaching DT</th>
<th>Using ICT course</th>
<th>Implementing DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.9</td>
<td>6.2</td>
</tr>
<tr>
<td>Median</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Mode</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Std. Dev</td>
<td>2.143</td>
<td>0.967</td>
</tr>
<tr>
<td>Min</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Max</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Before going into the course, participants reported that they were reasonably comfortable using ICTs in their everyday lives and activities (mean 6.2) but had limited experience teaching digital technologies (mean 3.9) and confidence implementing the learning area (mean 3.5).

Table 5: Survey participants’ previous experience with DT activities

<table>
<thead>
<tr>
<th>Previous experience with digital technologies</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>No previous experience</td>
<td>22</td>
</tr>
<tr>
<td>Visual Programming</td>
<td>12</td>
</tr>
<tr>
<td>Programming (general-purpose)</td>
<td>8</td>
</tr>
<tr>
<td>Other (basic computer use, internet, etc)</td>
<td>6</td>
</tr>
<tr>
<td>Microworlds LOGO</td>
<td>1</td>
</tr>
<tr>
<td>Robotics</td>
<td>1</td>
</tr>
<tr>
<td>Algorithm activities</td>
<td>1</td>
</tr>
<tr>
<td>3D simulations</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 5 presents participant experience with digital technologies lessons before starting the course. Many had no previous experience (22 responses), with some having had experience in using visual programming (12), general-purpose programming (8) and teaching students about basic computer, Internet and application use (6). Other items that mentioned were algorithm activities, 3D simulations, robotics and LOGO.

5.1.2 Participant experience in the course

A majority of participants reported completing all of the modules, which is consistent with our participation findings that those who completed the portfolio also completed all core and non-core modules. When survey participants were asked which modules they did not complete (6 in total), the following modules were identified in Table 6. Although some had not completed these modules they had intentions to return to view materials or, even if they did completed all modules, some still mentioned that they would revisit the modules in more detail. Others who had not tried visual programming during the course had set personal goals to learn visual programming in the near future.

Table 6: Survey participants’ ‘non completed’ modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data - Representation</td>
<td>1</td>
</tr>
<tr>
<td>Digital Systems</td>
<td>3</td>
</tr>
<tr>
<td>Information Systems</td>
<td>2</td>
</tr>
<tr>
<td>Algorithms &amp; Programming</td>
<td>1</td>
</tr>
<tr>
<td>Visual Programming</td>
<td>2</td>
</tr>
<tr>
<td>Portfolio</td>
<td>1</td>
</tr>
</tbody>
</table>

After the course, participant confidence to implement the new learning area had risen (mean 6.2) as well as confidence in implementing digital technologies lessons (mean 6.5) and making cross-curricula links (mean 6.2). However, we still have room to improve on increasing teacher confidence with digital technologies teaching strategies, the organization of content, designing activities, the integration of ICT and applying content knowledge. Survey participants were asked to select from a list of items, which activities they had tried for the first time since undertaking the course. Table 8 presents a frequency of the frequency of participants who reported trying the new activities.

Table 7: Survey participants’ confidence after the MOOC

<table>
<thead>
<tr>
<th>Statement</th>
<th>Mean</th>
<th>Mo</th>
<th>Std. Dev</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comfort Implementing DT</td>
<td>6.2</td>
<td>7</td>
<td>1.1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>AFTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidence Implementing</td>
<td>6.5</td>
<td>7</td>
<td>0.7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Lessons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applying Content</td>
<td>4.4</td>
<td>5</td>
<td>0.7</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Knowledge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organising Content</td>
<td>4.4</td>
<td>5</td>
<td>0.7</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Designing Activities for</td>
<td>4.3</td>
<td>5</td>
<td>0.7</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Survey participants’ experience in DT activities after the MOOC

<table>
<thead>
<tr>
<th>New activities tried</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm activities</td>
<td>24</td>
</tr>
<tr>
<td>Binary activities</td>
<td>23</td>
</tr>
<tr>
<td>Visual Programming</td>
<td>21</td>
</tr>
<tr>
<td>Other module topics</td>
<td>16</td>
</tr>
<tr>
<td>Data collection and analysis</td>
<td>13</td>
</tr>
<tr>
<td>Robotics</td>
<td>6</td>
</tr>
<tr>
<td>Code club</td>
<td>1</td>
</tr>
</tbody>
</table>

Previously 22 participants reported having had no experience with digital technologies activities or any of the items, but since participating in the course many had tried new activities for the first time. Many of the activities adopted for the first time were related to algorithms (24 responses), binary (23), visual programming (21), other module topics (16; digital systems, information systems) and data (13). Six participants had tried robotics and 1 person reported starting a school Code Club.

5.1.3 Perceived Challenges

Content analysis was applied to the participant responses to the questions about the challenges encountered by participating in the MOOC and the most challenging Modules. Time was a major factor identified as external pressures from work or personal life meant that they could not spend as much time on each module and activity as they would have liked.

Participants were asked to identify the most challenging modules. Participants reported that the later units from module 5 onwards (4 responses), algorithms and programming (9), binary (5) and other module 3 topics (3), and preparing the portfolio (3) were the most challenging. Two participants also mentioned that ‘everything’ was challenging. Using content analysis, we were able to group the reasons as to why participants felt that these topics or the MOOC in general was challenging. We identified a series of primary reasons that emerged relating to the content being challenging, transferring and applying knowledge, personal challenges and external factors.

The content was challenging: These reasons included that the topic was dry itself, that sometimes the information was overwhelming or too technical, or that they wanted to know more but were limited in time. A number of participants mentioned that the content was challenging because it was new (10 responses). Four participants mentioned that although the content was challenging it was exciting to learn and three mentioned that they intend to explore the course content further. Some interesting comments emerged around the language of the new curriculum – that once concepts, such as algorithms or iteration, were de-mystified they were far
more comfortable with the new curriculum and that the
terminology was less ‘scary’.

Transferring and applying knowledge: This topic included
reasons such as that designing lessons for this
new learning area was challenging (4 responses) and that
transferring knowledge to the classroom (3) or the design
of learning activities (4) for the community was a
challenge. One other participant who was in a leadership
role mentioned that transferring knowledge in the MOOC
for teachers would be a challenge for them because they
needed to understand it well enough themselves first at a
higher level to be able to transfer the knowledge.

Personal challenges: One participant expressed a
feeling of pressure to perform well in the community by
producing quality materials or lessons and another
participant expressed that they personally felt that they
struggled with a topic because they were not good at
mathematics. Although these only account for 2 out of 51
responses, others in the community may have also
encountered similar challenges.

Two external challenges were identified relating to
time constraints (32 responses) such as personal reasons,
workload, work pressures, life events that limited their
ability to participate or complete modules as well as
technical challenges (8), such as low internet speed,
computer issues and limitations imposed by school
contexts to access particular sites.

6 Conclusions
The expected changes in the teaching of CS represent a
significant challenge for our schooling systems. CT and
CS will form part of the Australian standard curriculum
from K-12 from 2014. In this paper, we have described
the CSER Digital Technologies MOOC, which supports
K-6 teachers in their development of CT awareness,
within the direct context of their learning and teaching
activities. We have described our course structure,
incorporating a specific example of how we have
focussed course activities within the teacher’s context,
incorporating a range of learning examples, with varied
tool dependency and integration across multiple existing
knowledge areas.

Our analysis has indicated that this course can assist
teachers in developing their understanding and
confidence in CT and digital technologies. Similar to
previous work that found weaving CT concepts throughout teacher courses was beneficial (Yadav et al 2011), we also found that unpacking core CT concepts (e.g. abstraction, algorithms, decomposition) and programming statements (e.g. functions, iteration) that featured in the curriculum, with everyday examples and cross-curriculum connections assisted teachers to understand and feel more comfortable the new curriculum. However, our cohort still indicated that the challenge of new content, and translation requirements for their immediate teaching context were still of concern, which is consistent with literature in the area. While we have provided one resource that addresses the required development of CT awareness, there is still substantial effort required not in providing needed resources, but also in further exploring appropriate pedagogy within the K-6 context. We identify that further research and development is required in building teaching strategies through exploring pedagogical research in K-6 digital
technologies education and translating effective pedagogy
to teachers through worked examples (e.g. pair
programming, teamwork, problem-based learning, etc).
Further, following teachers into the classroom to
determine impact of such PD courses in this field is
important.

Findings from the literature state that teachers suggest
computing education PD incorporate online community
networks to support teachers and facilitate the sharing of
resources (ACARA, 2013b; Black et al, 2013; Gander et
al, 2012). A core, and tentatively successful, aspect of
our course featured the development of a knowledge
sharing community; our future work seeks to evaluate the
community component and the more immediate and long-
term impact use of the community had on teacher support
and implementation.

7 References
ACARA. (2012): The shape of the Australian curriculum: technologies. Sydney, NSW: ACARA,
ACARA. (2013): The Australian curriculum: technologies information sheet. Sydney, NSW: ACARA,
Bagge, P (2012) Jam Sandwich Algorithm (programming
teacher bot), Computing At School (CAS), University
of Kent, available online, http://community.computingatschool.org.uk/resources/376
Bell, T., Curzon, P., Cutts, Q., Dagiene, V. & Haberman, B. (2011) 'Introducing students to computer
science with programmes that don't emphasise programming', Joint conference on Innovation and
technology in computer science education, Darmstadt, Germany.
Bell, T., Newton, H., Andreea, P., & Robins, A. (2012): The introduction of computer science to NZ high
schools: an analysis of student work. Workshop in Primary and Secondary Computing Education,
Hamburg, Germany, 5-15.
de Waard, I., Koutroupolos, A., Ozdamar Keskin, N.,
Abajian, S., Hogue, R., Rodriguez, C., et al. (2011) 'Exploring the MOOC format as a pedagogical
approach for mLearning', 10th World Conference on Mobile and Contextual Learning, Beijing, China.


Siemens, G. (2012), ‘MOOCs are really a platform’ ELearnSpace, [online] Available at: http://www.elearn.space/blog/2012/07/25/moocs-are-really-a-platform/


http://tunstalltimes.blogspot.com.au/2013/05/five-for-friday-holla.html