Constructionist Experiences in Teacher Professional Development: A Tale of Five Years

Daniel Hickmott, *daniel.hickmott@uon.edu.au*
School of Education, The University of Newcastle, Australia

Elena Prieto-Rodriguez, *elena.prieto@newcastle.edu.au*
School of Education, The University of Newcastle, Australia

Abstract

Computational thinking and coding have recently become compulsory elements in the Australian K-8 curriculum that should be taught using ‘authentic learning challenges’ (ACARA, 2018a). However, very few teachers, particularly in the primary school setting, have been schooled on computational thinking or coding and rarely possess pedagogies to teach them authentically. A range of professional development opportunities are currently being offered to impart this knowledge, both for content and pedagogy. In this paper, we provide an account of the evolution we have experienced when designing and improving professional development workshops for teachers in coding and computational thinking. We reflect on our challenges and successes, and attest that it was only after ‘discovering’ Constructionism in late 2015 that we have been able to prepare activities that truly emulate the authentic learning experiences that teachers are required to use in their classrooms.

Learning from pioneers: the influence of Seymour Papert on our professional development

Keywords

teacher professional development; constructionism; computing; programming

Introduction

Teacher preparation for new digital technologies curricula across the world has become a topic of interest in recent years. Globally, coding and computational thinking appear almost daily in the news, and there seems to be a strong sense within industry that they are skills essential for the workforce of the future (Chalmers & Quigley, 2017; Pappano, 2017).

In September 2015, the national Australian Curriculum was officially endorsed. This curriculum includes the new Digital Technologies (DT) subject, within the Technologies learning area, which is focussed on the teaching of *“computational thinking and information systems to define, design and implement digital solutions.”* (ACARA, 2018b). This subject will be mandatory for all Australian students from Kindergarten to Year 8 and available as an elective for Year 9 and 10 students.

Introducing a subject that has a focus on computing reflects a current global trend in K-12 Information and Communications Technology (ICT) education. Several countries have begun to change ICT curricula to focus less on the learning of particular software packages and to focus more on the teaching of skills core to computing, such as computational thinking and programming (Webb et al., 2016). It is argued that teaching students these skills will allow them to become creators, rather than just consumers of, digital technologies (Bower & Falkner, 2015).

Australian educational stakeholders have concerns about the feasibility of implementing the new DT subject. Falkner, Vivian, and Falkner (2014) revealed that *"a consultation with Industry, Community and Education stakeholders in Australia showed that 55% of respondents had concerns about manageability of the implementation of the curriculum, while 45% of respondents did not think that its learning objectives were realistic"* (p. 6). One of the stakeholders' main concerns is whether teachers are adequately prepared to teach computational thinking and programming because, unless they have qualifications in computing, they are unlikely to have had formal experience learning these skills. This is particularly true for primary school teachers, who don't usually have the option to complete a technology major and who are often generalist teachers (Vivian, Falkner, & Falkner, 2014).

Several teacher Professional Development (PD) initiatives have been developed by universities and private organisations to address concerns related to teacher preparation for the Australian DT subject (Commonwealth of Australia, 2016). These initiatives include Massive Online Open Courses (MOOCs), such as those run by the University of Adelaide's Computer Science Education Research (CSER) group (Vivian et al., 2014), and face-to-face workshops (Prieto-Rodriguez & Berretta, 2014).

Similar PD initiatives have also been developed in other parts of the world. In Europe, for example, a transnational initiative, TACCLE, combines learning through a website, news related to coding in schools, and the sharing of resource reviews by teachers (García-Peñalvo, 2016). In England, researchers, who were working with the Computing At School (CAS) organisation, developed a model of PD that is holistic and sustained for teachers implementing the Computing curriculum (Sentance, Humphreys, & Dorling, 2014). In the United States (US), computing PD research has mainly been conducted by computer science academics (Menekse, 2015). Menekse (2015) systematically reviewed US articles on computing PD published between 2004 and 2014. One of the aims of this review was to evaluate the effectiveness of these programs, according to five factors that the author determined to be indicators of effective PD from a literature review. The results of the review indicated that many of the face-to-face PD opportunities did not have these factors of effective PD present in them. Menekse suggested that one of the reasons why these factors were not present could be that these PD opportunities are often conducted by computer science faculty academics, who may have limited collaboration with other education researchers and practitioners.

In this paper, we reflect on our experience designing and improving PD for teachers in computing. We describe the learning process that we followed, as computer science academics new to computing education, trying to implement relevant approaches to PD that are valued by Australian teachers and useful in preparing them for the DT curriculum. Looking back, it is clear to us that it was after we started looking at our work through the lens of Constructionism, that we were able to make meaningful changes to our PD. The learning process we underwent to identify the changes to be made, highlighted the need to explore certain forms of PD that are not prevalent in recent computing education literature. We believe these forms of PD are not only useful for preparing existing teachers, but also for assisting in the development of pre-service teaching courses in computing. These courses should be designed so that they are relevant to current practice, particularly for primary school teachers.

Our Journey

Our team has been conducting computing PD workshops over the last five years. These workshops have been held at our university, conducted face-to-face, and have ran over two or three days. In 2013 and 2014, these were only available for high school teachers but in 2015 we began to include primary school teachers as well. Since their inception, the three overarching aims of the workshops have been to: 1) communicate the applicability and importance of computing to a wide range of research areas and careers, 2) provide examples of activities that address DT concepts and 3) provide resources for teachers to use in their classrooms. These workshops have involved computational thinking and programming exercises with step-by-step instructions, collaborative problem-solving exercises, and presentations by academics and industry representatives.

The design and implementation of the workshops has evolved each year as a result of participants' feedback, which has been collected through validated surveys (Prieto-Rodriguez & Berretta, 2014), review of the literature, and reflection. Initially, the changes were a direct consequence of the feedback received through the surveys, particularly within the open-ended questions. Later, since the 2016 workshops, changes were made to incorporate more constructionist activities.

When comparing the responses to the items common to all our surveys, it became clear that, as we included Constructionist approaches to the design of the workshops, the satisfaction with the content presented and its applicability to classroom practices increased. In particular the major changes introduced in 2016, as seen in Figure 1, seemed to increase the 'sense of community', 'inspiration to improve teaching' and 'use of applications of computer science', as we included more collaborative and hands-on activities (note: the asterisk next to the year indicates a primary school focus, all others are secondary school).

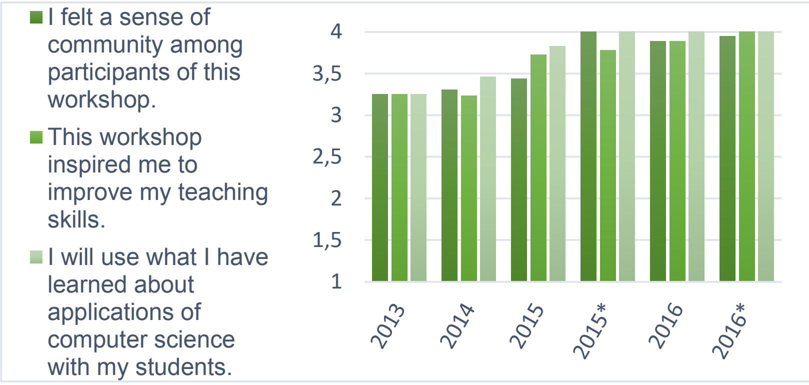

Figure 1. Mean responses to selected survey questions in the workshops delivered from 2013 to 2016. Items in a 4-point Likert scale with Strongly Disagree coded as 1 and Strongly Agree coded as 4.

Before Constructionism

The first PD workshop took place in 2013 and ran over three days. This workshop included a day of lectures presented by researchers from a variety of disciplines who incorporate computing in their work. The other two days involved a combination of presentations and some hands-on activities. Analysis of survey data of participants in 2013 showed that there were certain misconceptions with regards to the nature of computer science that we were able to unmask (XXX, 2014). In terms of the pedagogical approach utilised, feedback from the participants suggested that to improve future workshops we should include *"…more hands on with Robotics…"* and *"more hands on activities that we can take back and use in school*". It was clear that many of the teachers preferred to build their own knowledge rather than have it delivered to them in an instructionist manner.

In 2014, we increased the number of hands-on activities as a result of this feedback and reduced the workshop length to two days. Feedback from 2014 indicated that teachers wanted more ideas for projects that they could use with their students. One teacher suggested that we include *"Realistic projects for school students e.g. project ideas when using C++"* in future workshops. Reflecting back on these comments, we see now that teachers were looking for us to help them provide context and purpose to their students.

In 2015, we took a more systematic approach to the design of the PD and explored the use of student-centred learning approaches in the two two-day workshops that we ran that year. The hands-on activities included in the two previous years were written as step-by-step guides with explicit instruction on how to complete the activity. We adopted a framework for instruction based on constructivist principles to help us design and run collaborative problem-solving exercises. The Five Cs (5Cs) framework (Tom, 2015) is built on 5 constructs identified in the literature: *Consistency*, *Collaboration*, *Cognition*, *Conception*, and *Creativity*. The sessions in the two workshops were thus purposely designed to investigate participant responses to different pedagogies.

We used self-reported measures to compare perceptions about the activities designed with the 5Cs Framework against those that were step-by-step activities. Preliminary results were presented at the Constructionism 2016 conference (Prieto-Rodriguez & Hickmott, 2016). Our analysis showed that while concepts were understood consistently for all constructs, the Scratch session was the one were the teaching methods were preferred by participants and higher order thinking was more present. This session was the only one not involving collaboration and it was not designed using the 5Cs Framework.

The open-ended responses to the survey prompted us to reconsider what was valued by teachers and made us come to the realisation that higher order thinking and time to '*tinker'* should be a focus of our PD. This can be illustrated with one of the participant's survey responses stating that time to *'tinker'* would have been desirable: *"It would have been fantastic to have time to do a visual graphing experiment similar to the social media graphing presented on Tuesday morning. Use responses from the group to carry out an activity that could be done in the classroom".*

After Constructionism

In 2016, inspired by our attendance to the *Constructionism in Action* conference in Bangkok, we conducted two two-day workshops that incorporated Constructionist principles. The results from the 2016 surveys seemed to indicate that the inclusion of Constructionist approaches enhanced the satisfaction with all aspects of the workshop (see Figure). The workshops from that point thereafter, not only aimed to be more student-centred and include more hands-on experiences, but were also inspired by Constructionism.

As suggested by participants in the surveys, we had begun including an increasing number of hands-on activities in the workshops. However, it became apparent to us that it was important to incorporate hands-on activities that involved open-ended problem solving, in which teachers construct their knowledge in context, rather than just having hands-on activities that involved explicit step-by-step instruction.

To incorporate these Constructionist activities, we considered suggestions outlined by several Constructionist PD researchers and practitioners. For example, Martinez and Stager (2013) argue that PD is often *"too meta"* (p. 200), and suggest that teacher educators should provide PD where teachers experience learning from a student's perspective. Martinez and Stager designed and implemented *Constructing Modern Knowledge*, enabling teachers to spend four days of uninterrupted time working collaboratively on a project of their own creation. Brennan provides a similar argument concerning PD learning environments, stating, "teachers should have learning experiences that are comparable to their students' learning experiences, situated within a supportive community of fellow teachers" (p.293).

Brennan (2015) developed and implemented *ScratchEd,* a form of PD for computing that incorporated Constructionist principles. The main intent of *ScratchEd* is to support the use of technology for creating meaningful projects rather than focussing on the use of specific technologies. Like Martinez and Stager's *Constructing Modern Knowledge* workshops*, ScratchEd* is designed to involve experiences for teachers that comparable to their students' classroom experiences. Brennan*,* however, recognises that providing fully Constructionist learning environments can be at odds with K-12 mainstream education. She calls the difficulty of providing PD aligned with "the lived reality of K-12 education" (p.295) while preserving the ideal of a Constructionist environment, the "tension between the actual and the aspirational" (p.295). She also identifies four other tensions, which have guided and given focus to our reflections about the PD we provide.

We agreed with the sentiment that PD is often *"too meta"* (Martinez & Stager, 2013), and upon reflection, recognised that our previous PD opportunities had suffered from being too instructionist at times. However, like Brenan, when we introduced Constructionist learning into our PD, we had to negotiate some tensions.

In 2016, we ran two two-day workshops: a workshop for primary school (K-6) teachers and a workshop for high school (Years 7-12) teachers. These workshops involved activities and presentations that were aligned with the curriculum outcomes for the relevant school level. For example, in the primary school workshop, teachers took part in an activity where they used Scratch to learn about *visual programming*, which is a concept in the K-6 Australian Digital Technologies curriculum. In the 2016 workshops, in an attempt to move towards more Constructionist learning, we began to include some activities where teachers could choose their own direction for learning. In the high school workshop, for example, teachers could choose between learning about teaching Data Science with R or being introduced to general-purpose programming with Sonic Pi. Also, in the primary school workshop, we included extra activities for teachers who were already proficient at *visual programming*. Additionally, in the primary school workshop, we included a collaborative lesson planning activity, in which teachers worked together to construct a plan for introducing computing into their classes. However, we could not include longer self-directed learning, and had to negotiate with pressures such as the limited time available to teachers to attend the workshops, or the number of PD staff offering support to teachers working on individual projects.

In 2017, we revisited Brennan's tensions (2015) as a lens to improve the PD. The analysis of these tensions and our own reflections on how we negotiated them, were instrumental to envisioning the changes we made to the workshops in 2017. One of the main changes introduced that year, which responded to our reflection on the *tension between the actual and the aspirational* identified by Brennan (2015), was the introduction of workshops addressing specific curriculum outcomes.

In light of this, we ran two new specific workshops, one for primary teachers and one for high school mathematics teachers. These two workshops aligned entirely with areas of the curriculum and were designed to use computing as an exploratory tool for teachers (and subsequently their students) with mathematics. The first of these workshops utilised one of the modules produced for the *ScratchMaths* project (Benton, Hoyles, Kalas, & Noss, 2017). We mapped the content of the module to the Australian curriculum. The second workshop targeted a new area of the Year 12 curriculum, Minimum Spanning Tree algorithms. In this workshop, teachers learned and used *Edgy*, an adaptation of *Snap!* for graph theoretical explorations (Cox, Bird, & Meyer, 2017). Teachers coded Kruskal and Prim's algorithms with varying degrees of guidance after reviewing other content, such as graph theory and data structures, necessary for the completion of the task.

We also ran two more two-day workshops, similar to the ones in 2016 but more general in scope, and deliberately targeting outcomes, such as *critical and creative thinking*, from the General Capabilities of the Australian National Curriculum (ACARA, 2018b). These two workshops specifically addressed issues of differentiation of learning and learners - *the tension between novice and expert* identified by Brennan (2015).

In workshops prior to 2017, we had assumed that the teachers attending were novice to computing, and we felt that most of the PD content had to be heavily scaffolded. However, teachers with expertise in computing also attended the workshops. To better support these expert teachers, thus helping us negotiate this last tension, we interviewed a local teacher who had participated in our workshops in 2013, and had been invited to present in subsequent years' workshops. This teacher is a high school DT teacher with an academic and professional background in computing. She is experienced in teaching computing, regularly presents at conferences and assists in training her colleagues and thus fits in the category of *Master Teacher* in the classification devised by Sentance et al. (2014). Master Teachers are experienced educators chosen by CAS in England to organise and run face-to-face workshops with teachers in their local community. We asked questions to ascertain the extent to which her approach to teaching and learning was a constructionist one:

Question: *"Do you learn using YouTube, read instructions, or tinker?"*

Answer: *"Yes! First place I go is YouTube and tinkering is the best part of learning and teaching."*

Question: *"Do you use learning resources created specifically for teachers? If so, what were the most helpful aspects of these sort of experiences?"*

Answer: *"Sometimes but these resources often lack room for creativity or make teachers into dependant zombies instead of content creators. However these can be great for networking and idea generation."*

We also asked the teacher to provide insights into their learning for the classroom. We found that tinkering is also a part of the process to develop their pedagogy.

Question: *"Do you find that the activities you find need to be modified to suit your needs? If so, which type of activities and how do you modify them?"*

Answer: *"Most often, yes. Usually tech PD or resources are aimed at beginners so I often extend or find additional bits to meet my needs or particular learning requirements for my students."*

Her responses to these questions quite clearly indicate that her approach to both learning and teaching is a Constructionist one, and that she is at a level were step by step instruction is no longer necessary. One of the foci of our future PD endeavours will be to incorporate more differentiation, which would allow expert teachers to learn during our PD while respecting that other teachers need more guidance.

Discussion and Future Work

The recently endorsed Australian National Curriculum includes General Capabilities (such as *critical and creative thinking*), which teachers are expected to assess and report across all subject areas. In addition, this curriculum includes computational thinking and programming as part of the new DT subject that should be taught using 'authentic learning challenges' (ACARA, 2018a). These changes in the Australian curriculum provide an ideal opportunity for teachers to include Constructionist activities in K-12, which involve computing and open-ended problem solving. As argued by Angeli et al. (2016), there are many opportunities for K-12 educators to include authentic learning challenges that incorporate computing and also address outcomes from diverse content areas. In Australia, these authentic learning challenges could be particularly beneficial to K-6 teachers, who often teach multiple content areas, as they could target outcomes from a variety of subjects and also address the General Capabilities.

Due to the large number of teachers that need to be upskilled in computing, there has been a focus on creating computing PD that is scalable. An example of PD designed to be scalable that has been very successful, are the MOOCs designed by CSER. These MOOCs have reached many teachers, both nationally and internationally (Falkner, Vivian, Falkner, & Williams, 2017; Vivian et al., 2014), and CSER received government funding for a national rollout of their initiative, focussing on teachers in remote and disadvantaged areas (Birmingham, 2016).

There are many advantages of providing PD through online courses. Online courses such as CSER's MOOCs, are useful scalable solutions, as teachers can access them on their own time and at their own pace, and they are generally free of cost. However, as Brennan (2015) argues, it can be difficult to provide Constructionist learning experiences solely through an online course. As PD providers who have been influenced by Constructionist ideas, we believe that the inclusion of authentic learning challenges in K-12 should be encouraged. Thus, we reflect that our PD should include face-to-face components and authentic learning challenges to prepare teachers effectively. This reflection has led us to make further changes to our PD which we plan to implement in 2018.

In 2018 we will run three PD programs. The first program will be a two-day workshop for 'novices', with a strong emphasis on the Australian General Capabilities and will incorporate authentic tasks that we will develop with the teachers. Our second PD program will focus on expert teachers who are interested in running professional learning events for other teachers at their school or in their professional learning communities. Like the first workshop, it will involve hands-on activities where teachers will familiarise themselves with the content to be delivered, but they will also learn to run workshops for colleagues. We will facilitate a session with the aim of teaching about the logistics of running this type of event. We believe that this scalable model can complement online learning platforms whilst providing opportunities for Constructionist learning.

Our third PD program is focussed on sustained engagement and will run over six months with primary school teachers. This program will incorporate a greater depth of Constructionist experiences for teachers. Finding ways to provide such PD is the main aim of our future research, and constitutes the core research of Author 1's PhD studies. As well as improving PD, these studies could inform the design of learning experiences for pre-service teachers in computing, which could help address pipeline issues (Yadav, Sands, Good, & Lishinki, 2018).

We believe that the insight gained from designing, reflecting on, and researching PD has not only improved our PD programs, but also has the potential to create Constructionist learning opportunities for pre-service teachers that will ensure that they are able to include authentic learning challenges in their future teaching.

Acknowledgements

The authors would like to thank Google Inc. for funding the CS4HS workshops held at our institution. We would also like to thank the university academics who presented talks and gave laboratory tours during the workshops, as well as the administrative staff of our institution.

References

ACARA. (2018a). Digital Technologies curriculum rationale. Retrieved from https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/rationale/

ACARA. (2018b). F-10 Curriculum - General Capabilities. Retrieved from https://www.australiancurriculum.edu.au/f-10-curriculum/general-capabilities/

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: implications for teacher knowledge. *Journal of Educational Technology & Society, 19*(3), 47.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: some findings of design research in England. *Digital Experiences in Mathematics Education*.

Birmingham, S. S. (2016). Closing the 'digital divide' for disadvantaged students [Press release]. Retrieved from http://www.senatorbirmingham.com.au/Latest-News/ID/2928/Closing-the-digital-divide-for-disadvantaged-students

Bower, M., & Falkner, K. (2015). *Computational Thinking , the Notional Machine , Pre-service Teachers , and Research Opportunities*. Paper presented at the Australasian Computing Education Conference, Sydney, Australia.

Brennan, K. (2015). Beyond Technocentrism: Supporting Constructionism in the Classroom. *Constructivist Foundations, 10*(3), 289-296.

Chalmers, J., & Quigley, M. (2017, 13 October). Twenty ideas for our schools and politicians in the new machine age. *The Sydney Morning Herald*. Retrieved from https://www.smh.com.au/technology/twenty-ideas-for-our-schools-and-politicians-in-the-new-machine-age-20171012-gyzfm7.html

Commonwealth of Australia. (2016). *STEM Programme Index*. Retrieved from http://www.chiefscientist.gov.au/2016/01/spi-2016-stem-programme-index-2016-2/

Cox, R., Bird, S., & Meyer, B. (2017). *Teaching Computer Science in the Victorian Certificate of Education: A Pilot Study.* Paper presented at the Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.

Falkner, K., Vivian, R., & Falkner, N. (2014). *The Australian Digital Technologies Curriculum: Challenge and Opportunity*. Paper presented at the Australasian Computing Education, Auckland, New Zealand.

Falkner, K., Vivian, R., Falkner, N., & Williams, S.-A. (2017). *Reflecting on Three Offerings of a Community-Centric MOOC for K-6 Computer Science Teachers*. Paper presented at the Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, Washington, USA.

García-Peñalvo, F. J. (2016). *A brief introduction to TACCLE 3—coding European project.* Paper presented at the International Symposium on Computers in Education (SIIE), Salamanca, Spain.

Martinez, S. L., & Stager, G. (2013). *Invent to learn: Making, tinkering, and engineering in the classroom*: Constructing modern knowledge press Torrance, CA.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education, 25*(4), 325-350.

Pappano, L. (2017, 4 April). Learning to Think Like a Computer. *The New York Times*. Retrieved from https://www.nytimes.com/2017/04/04/education/edlife/teaching-students-computer-code.html

Prieto-Rodriguez, E., & Berretta, R. (2014). *Digital technology teachers' perceptions of computer science: It is not all about programming.* Paper presented at the Frontiers in Education Conference (FIE2014), Madrid, Spain.

Prieto-Rodriguez, E., & Hickmott, D. (2016). *Preparing teachers for the Digital Technologies curriculum: preliminary results of a pilot study*. Paper presented at the Constructionism 2016 Conference, Bangkok, Thailand.

Sentance, S., Humphreys, S., & Dorling, M. (2014). *The network of teaching excellence in computer science and master teachers*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education, Berlin, Germany.

Tom, M. (2015). Five C Framework : A student-centered approach for teaching programming courses to students with diverse disciplinary background. *Journal of Learning Design, 8*, 21-37.

Vivian, R., Falkner, K., & Falkner, N. (2014). Addressing the challenges of a new digital technologies curriculum: MOOCs as a scalable solution for teacher professional development. *Research in Learning Technology, 22*.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, 1-24.

Yadav, A., Sands, P., Good, J., & Lishinki, A. (2018). Computer Science and Computational Thinking in the Curriculum: Research and Practice. In J. Voogt, G. Knezek, R. Christensen, & K.-W. Lai (Eds.), *Handbook of Information Technology in Primary and Secondary Education* (pp. 1-18). Cham: Springer International Publishing.