Beyond lesson recipes: first steps towards a repertoire for teaching primary computing

Chris Shelton, *c.shelton@chi.ac.uk*
Institute of Education, University of Chichester

Abstract

In 2014, the UK government introduced a new National Curriculum for state schools in England with a greater emphasis on computer science and computational thinking. Teaching this new curriculum presented challenges to many primary school teachers and led to a demand for professional development and exemplar teaching resources. This paper argues that many of the resources created in response to the revised curriculum are 'recipes' for lessons that fail to prepare teachers to teach challenging and purposeful computing lessons. It argues that, instead of providing recipes, we need to develop teachers' 'repertoire' of strategies for teaching computing and that our approach to doing this should take account of the context in which primary teachers now work.

The paper describes professional development practices designed to help less confident teachers take their first steps away from model lessons and towards computing projects that reflect the needs and interests of the pupils they teach.

In particular, this paper will focus on two aspects of these practices: a teaching sequence intended to scaffold teachers in planning and teaching computing, and an approach to meeting the needs of the range of learners in a primary classroom through self-directed challenges. These were intended to support primary school teachers in improving their confidence and capability to plan and teach computer programming.

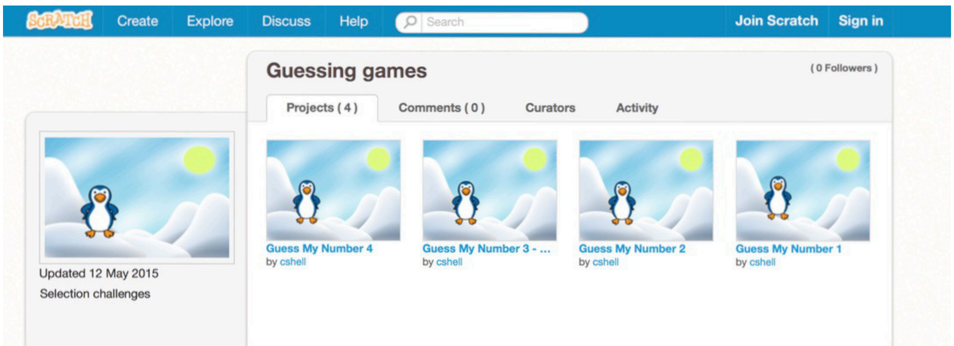

Figure 1. Scratch studio of learning challenges

Keywords:

professional development, computing, computational thinking, primary schools, elementary schools, pedagogy

Introduction

This paper describes professional development practices designed to develop primary school teachers' confidence and capability to plan and teach computer programming. Within the constructionist community, there is a long tradition of very successful approaches to professional development for teachers. In the UK, however, recent changes to the National Curriculum for England and the wide availability of new example teaching materials have changed the landscape for primary computing and led to new demands for teacher professional development. This paper discusses an approach to support teachers to plan purposeful computing activities, including those teachers who are risk-averse or lack the confidence to step outside familiar and 'safe' teaching approaches.

As this paper claims that the context in which teachers are situated should shape our approach to professional development, it will first set out the context for primary computing in England. It will then discuss two aspects of professional development: a teaching sequence intended to scaffold teachers in planning and teaching computing, and an approach to meeting the needs of the range of learners in a primary classroom through self-directed challenges.

Context and background

In 2014, the UK government introduced a new National Curriculum for state schools in England (DfE 2013). One of the biggest departures from the previous curriculum (that had been in place since 2000) was that the subject of Information and Communication Technology (ICT) was renamed 'Computing' and the content of this subject was revised with a much greater emphasis on computer science and computational thinking. These changes presented challenges for teachers of primary pupils (ages 5-11), the majority of whom are non-specialists with limited or no experience of computer science.

Unsurprisingly, teachers' lack of subject knowledge alongside a curriculum that introduced complicated and unfamiliar vocabulary posed a challenge to the introduction of the new curriculum (though arguably, this vocabulary could be seen as fairly uncomplicated concepts disguised in technical language). While there is a strong tradition of high quality computing in the UK, most notably, the teaching of LOGO enthusiasts and a small core of schools where children achieved much more than the requirements of the National Curriculum, achievement in primary schools prior to 2014 varied widely. According to Ofsted (2011), in nearly two thirds of primary schools the teaching of ICT was good or outstanding but there were particular weaknesses in the teaching of programming or 'control' technology.

Solutions to this challenge have been numerous. The UK Department for Education provided some funding for professional development for teachers notably through a website: 'Barefoot' computing (http://barefootcas.org.uk) and the Computing At School (CAS) subject association. In parallel to this, initiatives such as the 'Hour of Code', code clubs, and commercial publishers of teaching materials offered an increasingly wide choice of resources to teachers. This has included online 'coding' puzzles, lesson plans for lessons using computers and 'unplugged' activities that teach principles of computer science away from the computer itself. Such resources offer teachers 'easy wins' in their high pressured and busy schedules. At a local and regional level, there has been a rise in demand for continued professional development as schools search for teachers with expertise to offer training to their schools. Many of these professional development events have needed to concentrate on building teachers' confidence and providing reassurance. As teachers have developed their subject knowledge and been able to relate the vocabulary of the curriculum (computational thinking, algorithm, selection, etc.) to their practice they have grown in confidence and their initial fears have been quelled. As a result, they have adopted some of the example lesson ideas they have been given and started to teach computing to their pupils.

However, although these resources and basic computing skills training has provided an initial solution to teachers' anxieties about what to do in their new 'computing' lessons, they fail to equip teachers to be able to design the rich, open-ended learning opportunities that will most benefit their pupils and that are a feature of the practice of constructionist teachers.

In fact, many of the teaching resources provided in response to the demands of the 2014 curriculum are best understood as lesson 'recipes'. Alexander (2010) draws a distinction between teaching 'recipes' and teaching 'repertoire'. While lesson ideas and exemplar plans can provide a lesson recipe for teachers to follow, and this can be exactly what an anxious and inexperienced teacher desires, for a teacher to be able to design activities that meet the needs of the pupils they teach, they need to have a repertoire of teaching strategies that they can use and adapt based on their professional judgement. However, it can be very challenging for teachers to have the confidence to step away from model lessons designed by 'experts' and to take full control of their lessons even though classroom teachers are best placed to know the individuals they teach and to make connections between the required curriculum and the interests and aptitudes of their pupils. Alexander suggests that in primary education, educators need to "work towards a pedagogy of repertoire rather than recipe, and of principle rather than prescription" (2010, p511). This is particularly the case for computing in England at this time. As the initial shock of a new curriculum and the new language of computer science begin to fade, English primary teachers need support to move beyond the recipes they have been using in the first year of the computing curriculum and to develop a teaching repertoire that will support them and their pupils to achieve much more than the National Curriculum requires. In addition, it is a grounding in principles of good practice (and, in particular, constructionist principles) that can help to provide teachers with the confidence to do this.

This paper discusses some of the practices that have developed over the last year of attempting to develop primary teachers' computing repertoires in the new context they find themselves. The principal aspect of the approach described here has been to present a particular teaching sequence that can be seen as a first strategy to adopt. It is envisaged that this sequence, far from being the only way to teach computing, is a first step towards building a teachers' repertoire: a single strategy but one that can be applied in many lessons and which lends itself to being adapted for different topics. At its heart is the assumption that children will learn computing best through the creation of meaningful projects rather than limited puzzles (Resnick, 2014). The second aspect of this approach is to help teachers to learn to meet the needs of the learners in their class through self-directed challenges.

In addition, this paper will make explicit the connections between these teaching approaches and two other concerns of primary teachers: dialogue and 'mindset'. This serves to help teachers to connect what they need to do when teaching computing to their understanding of good practice in the rest of their teaching, specifically, the importance of dialogic teaching (Alexander, 2006) and the benefits of encouraging a 'growth mindset' (Dweck, 2006). These are increasingly becoming accepted ideas amongst UK educators and as such, provide a familiar anchor from which to start improving computing teaching.

A teaching sequence for primary computing: "UpTIME"

Soon after the launch of 2014 National Curriculum, it became clear that the initial professional development focus on teachers' programming or 'coding' skills and knowledge of computer science concepts and vocabulary, whilst important, was not going to give teachers the pedagogic skills necessary to move beyond the lesson 'recipes' that had suddenly become popular in primary schools. Therefore, the challenge for teacher professional development was to encourage and support teachers to take greater control of their planning and teaching for computing. While some teachers were quick to appreciate that the language of the National Curriculum can be interpreted very broadly and that it provides scope for varied and exciting lessons, others needed much more support and reassurance.

During 2014-15, staff at the University of Chichester were asked to run a sequence of computing workshops for pupils from local primary schools. These focused on using Scratch to meet and exceed the requirements of the National Curriculum (and also on empowering pupils to return to their schools and support others who were learning computing). In discussing with teachers how these sessions were designed and the principles underpinning them, it was clear that an analysis of these workshops could provide a template for future planning. This template became the teaching sequence described below. The intention of this work was not to provide another lesson 'recipe' but rather to provide a scaffold – a structure and strategy for planning that could be applied across many aspects of the computing curriculum. Secondly, the sequence was designed in such a way that by introducing the sequence, both new and experienced teachers would be made aware of some of the principles that underpin computing teaching.

A single page summary of the teaching sequence written for teachers and intended to be used in professional development sessions can be found online at:

https://challengingcomputing.wordpress.com/uptime/

The teaching sequence was given the acronym "UpTIME" which stands for:

- Use/ play
- Tinker
- Improve
- Make
- Evaluate

It is designed as a simple way to plan effective sequences of activities to teach computing that teachers can use or refer to when planning. There is intentionally no indication of how long teachers should spend teaching the sequence or on any individual part of the sequence and, in fact, teachers are encouraged to consider how it might be used within a single lesson or over a much longer series of lessons.

For experienced teachers, one of the most exciting things about project-based learning in computing is that children will surprise us by developing ideas and products that we could not have imagined at the start of a project and that 'teachable moments' present themselves when least expected. However, for the teachers for whom this teaching sequence is intended, this can be a daunting prospect. Therefore, the sequence encourages teachers to identify a specific aspect of computing that they wish their pupils to learn based on their prior assessment of pupils' learning. It is intended as a 'learning driven' rather than 'activity driven' approach which complements the pedagogic practices that they are familiar with from English primary teaching.

U/p – USE / play

The first stage of the sequence is intended to allow teachers to provide an authentic and purposeful context for the computing concepts and skills that they want their pupils to learn. It encourages teachers to allow their pupils to use programs that make use of the specific elements of computing being taught and, if this program is in the form of a game, then to play the game. For example, if the teacher has identified that their pupils need to develop their understanding of variables then they might play a game that makes use of variables in a number of different ways, e.g. for scores, lives or levels.

Some of the lesson 'recipes' used in UK schools begin by introducing abstract concepts or vocabulary (often away from the computer). While there can be a place for this, teachers need to motivate and enthuse their class and help them to see the relevance of what they are being taught. Playing games that use the concepts that will be introduced later is one way of providing a context for later work. Alternatively, sometimes the initial program that children use or play may be incomplete or 'broken' to draw their attention to the part of the program that has gone wrong.

One aim of the sequence is to maximise opportunities for children to learn through productive talk with peers and adults so at this stage teachers can be encouraged to ensure that pupils try to explain to a partner or the class how they think the game works.

T – TINKER

The second stage in the sequence asks teachers to let pupils investigate the particular learning focus by tinkering with the program or game that they have already used. This involves pupils looking closely at the program, discussing with their partners how it works, changing the program and seeing what happens. Papert (1993) described learning itself in terms of tinkering or 'bricolage': "building up a set of materials and tools that one can handle and manipulate" (p173) and here the pupils are learning through their manipulation of the program.

This is also an opportunity for the teacher to encourage 'dialogic' talk that is cumulative, reciprocal, supportive and purposeful (Alexander, 2006). They can share their pupils' 'tinkerings' with the rest of the class and let others build on these ideas. They can also elicit key concepts and model the vocabulary they want their pupils to use.

I – IMPROVE

As soon as pupils start tinkering, they discover ways to 'break' programs and ways to improve them. At this stage, teachers are encouraged to allow the children to make small purposeful changes to programs. Some of the ideas for improvement will come from the children while others may be designed by the teacher. It is at this stage that teachers are encouraged to cater for the range of abilities that can be found in their class through setting a variety of different learning challenges (see below).

M – MAKE

This stage is the most crucial and the teaching sequence is designed to scaffold teachers towards this. By giving children the opportunity to design and make their own projects, they develop independence, are motivated to learn new concepts to achieve specific outcomes, and enabled to make connections between abstract concepts and their application. Here, teachers can support pupils to design programs, decompose them into manageable tasks, create, debug and share their work.

However, by placing these open-ended (and potentially daunting) projects within a focussed series of activities, the sequence supports teachers in two ways. Firstly, by helping them to connect such projects to the specific demands of the National Curriculum and, secondly, by allowing them to gain confidence through fairly tightly defined activities before 'letting go' and enabling their pupils to explore more widely.

E – EVALUATE

The final part of the teaching sequence is 'evaluate' but this is not meant to imply that pupils only evaluate their work (both their learning and their programs) at the end of the sequence. Children will evaluate their work at every stage of this teaching sequence: they will evaluate other peoples' programs at the start of the sequence and they will improve and debug their own programs continually when programming. But, in addition to this, teachers need to create space in their timetable for individuals and groups of children to reflect upon and evaluate the technologies they have used, their own creations, their skills and their learning.

Setting suitable learning challenges

If there is one event that most clearly demonstrates the limitations of model lessons and teaching 'recipes', then it is when these lessons fail to meet the needs of many of the children in a group. In any primary classroom, the pupils will have a wide range of attainment in computing. While some children may have had limited access to technology outside of school, others may use technology frequently at home and enjoy to explore and creative with digital tools. As a result, lesson ‘recipes’ designed for an imaginary class of children can frequently fail to meet the needs of the most or least confident pupils.

In the UK, primary classes are rarely set or streamed and teachers are experienced in catering for a wide range of attainment in one class. However, media images of children as expert computer users (cf. Selwyn 2003) can lead teachers to view their pupils as very likely to be much more knowledgeable about computing than they are and this can lead to anxiety about teaching computing.

As a result of this concern, a second aspect of professional development for primary teachers has been to demonstrate and promote an approach to meeting the needs of the range of learners in a primary classroom through self-directed challenges. This is intended to help teachers ensure that all their pupils are suitably challenged, in particular that higher attaining pupils are extended but that those who might struggle with computer programming do not become frustrated and disengaged.

As mentioned above, one of the aims of this approach is to make connections between the teaching of computing and primary teachers’ understanding of pedagogy in other subjects that they may be more confident teaching. In this particular case, it is useful to introduce the approach with reference to Dweck’s (2006) work on ‘growth mindsets’ and the application of this to the primary classroom through the ‘Learning without Limits’ project (Hart *et al*., 2004).

Carol Dweck’s work on ‘growth mindsets’ has grown in popularity in the UK and increasingly teachers are becoming aware of the difficulties caused when teachers view pupils as being of fixed ability and when pupils view themselves this way. In addition, Hart *et al.* (2004) have demonstrated how the practice, commonly found in England, of classifying children as ‘high ability’ or ‘low ability’ learners can lead to unhelpful fixed mindsets.

If we apply these ideas to teaching computing, then those pupils labelled as ‘low ability’ may believe that they will never be able to master computing while those labelled as ‘high ability’ may become reluctant to take risks for fear of making mistakes – a particular problem for a subject like computing that has ‘tinkering’ and learning through error at its heart.

One of the strategies used by schools in the ‘Learning without Limits’ project to encourage children to fully participate and take responsibility for their learning was the use of learning challenges. In applying this to the teaching of computing and the demands of the National Curriculum, teachers are encouraged to set three or four challenges for their pupils and to allow the children to select the most appropriate challenge for themselves. Pupils are not required to work through all of the challenges in order but are encouraged to take responsibility for their learning.

For example, at Key Stage Two of the National Curriculum for England (Key Stage Two refers to pupils aged 7-11), pupils are required to “work with variables”. If a teacher is using the ‘UpTIME’ sequence, then the children might begin by playing games that include variables for scores, levels, etc. Then they will tinker to discover how these variables work and the teacher will lead the class in exploring and discussing the concept of variables and how (and why) they are used. The tinkering stage should be quite open-ended so that pupils have some freedom to explore for themselves but this should not overwhelm anyone so the teacher might set some ‘tinkering challenges’. Then he/she might ask the children to improve the games and could set simple challenges (e.g. change how many points are scored each time) or more complex ones (e.g. add a new variable to represent how many ‘lives’ the player has left and make the game finish if they lose all their lives).

If the teachers are using Scratch then studios can be very effective ways of sharing the set of challenges. Below is another example – a Scratch Studio of challenges called ‘Guessing Games’ (see Figure 1).

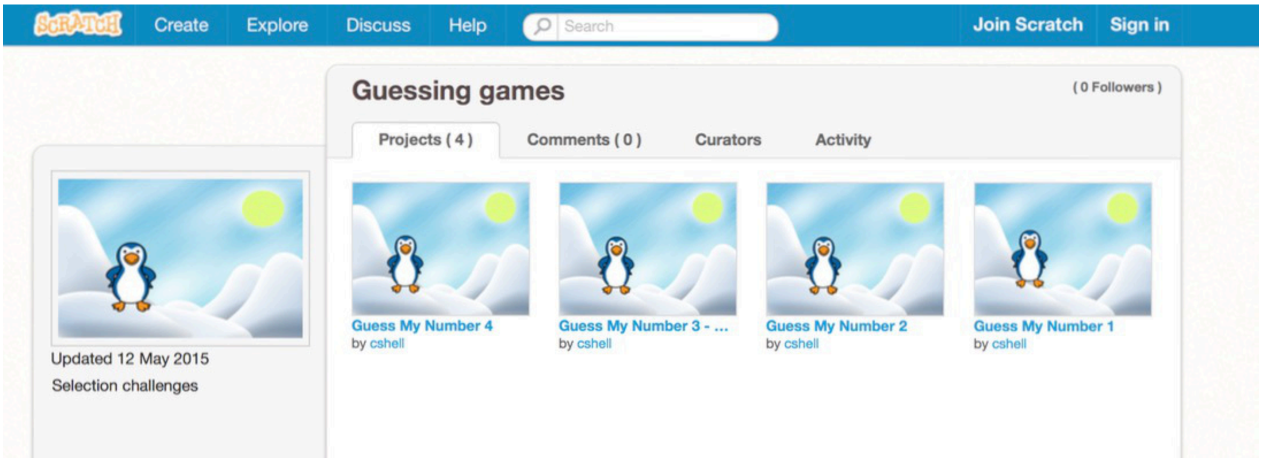

Figure 1: Guessing Games Studio

This studio was designed to help teach selection using 'if' statements. The four different Scratch projects all use the 'if' or 'if…else' block but get progressively more complex. Guess My Number 1 is the simplest program and allows pupils to focus on how the 'if' condition works. The second version improves the guessing game but is more complicated (for example, the 'if' block is nested within another block). The third game shows how the game can work with words as well as numbers while the fourth introduces 'more than' and 'less than' operators. The four games can be used to challenge pupils in several different ways. For example, a teacher might ask his/her pupils to choose one of the games and explain how it works to a partner, or they might ask pupils to improve one of the games and let them pick one that they feel they understand as their starting point. For a different class, a teacher might need to give the children more challenging opportunities, for example, a challenge to create a game that uses random numbers or that keeps a count of how many attempts it takes a user to guess the correct answer.

In summary, encouraging teachers to set a range of learning challenges in computing gives them a strategy to meet the needs of the range of learners in their class while also helping to make connections between this 'new' subject of computing and their existing understandings of effective primary pedagogy.

Conclusion

Faced with the multiple challenges of life as a teacher in the 21st century, it is no surprise that educators rush to collect teaching 'recipes' that offer easy solutions to planning computing. But while the limitations of such recipes are well known, finding accessible ways of moving from recipes to a varied teaching repertoire is challenging. While we know that enthusiastic teachers will find ways of navigating through curriculum documents and teach in innovative ways, more support is needed to help less confident teachers.

The teaching sequence offered here is not a recipe to teach a single lesson but rather a tool for teachers to use and adapt to enable them to meet the needs of their pupils. The teaching sequence described in this paper might be thought of as a scaffold to support teachers as they move from following recipes to designing purposeful and authentic teaching and learning experiences. It is not intended as an end in itself but rather as a first step in the direction of a wide constructionist teaching repertoire.

A number of questions remain about how teachers develop a full and varied pedagogic repertoire for computing that require further empirical investigation. We are beginning to explore how the teaching sequence described here is used by teachers and what other approaches might function as first steps to encouraging teachers who are confident enough to abandon their recipe books and design their own teaching and learning opportunities. In addition, while there is excellent evidence for the use of dialogic teaching that uses digital technology to enhance learning across the curriculum (e.g. Wegerif and Dawes, 2004), more research is needed to investigate how teachers can be supported to model and promote exploratory talk when teaching computing. Similarly, while there is excellent work on growth mindsets in certain other subjects, there are opportunities to do more to investigate how these concepts relate to the teaching of computing.

References

Alexander, R. (2006). *Towards dialogic teaching: Rethinking classroom talk.* Dialogos, York.

Ed. Alexander, R. (2010). *Children, their World, their Education: final report and recommendations of the Cambridge Primary Review.* Routledge, Abingdon.

Department for Education (2013) *The national curriculum in England: Key stages 1 and 2 framework document*. Available at: https://www.gov.uk/government/publications/national-curriculum-in-england-primary-curriculum [Accessed 14 October 2015]

Dweck, C. (2006). *Mindset: The new psychology of success.* Random House, New York.

Hart, S., Dixon, A., Drummond, M.J. and McIntyre, D. (2004) *Learning without Limits.* Maidenhead: Open University Press

Ofsted (2011) *ICT in schools 2008–11: An evaluation of information and communication technology education in schools in England 2008–11.* Available at: https://www.gov.uk/government/publications/ict-in-schools-2008-to-2011 [Accessed 14 October 2015]

Papert, S. (1993). *Mindstorms: Children, Computers, And Powerful Ideas (Second Edition).* Basic Books, New York.

Resnick, M (2014) *Give P's A Chance: Projects, Peers, Passion, Play.* Paper presented at Constructionism 14 Conference, August 19th - 23rd 2014, Vienna, Austria. Available at: http://constructionism2014.ifs.tuwien.ac.at/papers/1.2_1-8527.pdf [Accessed 14 October 2015]

Selwyn, N. (2003). *Doing IT for the Kids': Re-examining Children, Computers and the Information Society'*. Media, Culture & Society, 25(3), 351-378.

Wegerif, R., Dawes, L. (2004) *Thinking and Learning with ICT: Raising Achievement in Primary Classrooms*. Routledge, London.