Paper presented at IFIP TCS 2015 – June 2015

Teachers' perspectives on successful strategies for teaching Computing in school

Sue Sentance, *sue.sentance@kcl.ac.uk*
King's College London, London, UK

Andrew Csizmadia, *a.p.csizmadia@newman.ac.uk*
Newman University, Birmingham, UK

Abstract
With the introduction of Computing into the school curriculum in England, experienced teachers are having to teach it for the first time. This raises a number of questions, including whether teachers need to adapt their existing pedagogical strategies to deliver Computing in the curriculum. In this paper we address this particular question through an analysis of qualitative statements made about how to teach Computing by over 300 in-service teachers, who contributed as part of a larger survey. We identify a range of pedagogical strategies that teachers use in practice which can be categorised into the five areas of contextualised learning, computational thinking skills development, code manipulation, working collaboratively and learning away from the computer. We suggest that focusing on the use of a range of these strategies could help teachers to feel more confident in the Computing classroom.

Keywords
Computer science education, Pedagogy, Primary computing education, Secondary computing education, Teachers' perspectives, Computational thinking

INTRODUCTION
Computing is being introduced as a new subject in the school curriculum in many countries, and as an important part of informal learning opportunities in others. This brings with it both excitement and challenges, as for any new subject. For teachers facing curriculum change, how to teach it is very pertinent. Introducing new content does not merely mean that teachers have to equip themselves with new subject knowledge, which of course in many cases they do (Brown et al., 2013; Sentance, Dorling, & McNicol, 2013; Thompson & Bell, 2013). Teachers also need to learn appropriate pedagogies for delivering a new subject, particularly in those aspects of computer science that relate to algorithms, programming and the development of computational thinking skills.

In this paper, statements made by teachers who are currently teaching Computing in primary and secondary schools have been coded, categorised and analysed, describing both successful strategies for teaching and the difficulties they face. The teachers' perspective gives us some interesting evidence of what works for real teachers in their classrooms.

TEACHING COMPUTING IN SCHOOL
Recent literature relating to computer science education in school highlights a number of ways of making computer science concepts accessible, engaging and fun, and more importantly, giving learners a deep understanding of these concepts.

Constructivist theory, based on the work of Dewey (1938), Piaget (1950) and Bruner (1996) suggests that learning is a cumulative and active process during which the learner constructs knowledge and meaning for themselves as they learn, connecting with, and explaining new knowledge in terms of, what they already know. Constructivist learning theories applied to computer science emphasize the active, subjective and constructive character of knowledge, placing students at the centre of the learning process (Ben-Ari, 1998). Specifically, constructivist learning, based on students' active participation in problem-solving and critical thinking, has profoundly influenced the teaching of programming (Ben-Ari, 1998).

Experiential learning that stems from constructivism describes the design of activities which engage learners in a very direct way. Working with tangible real world objects is a central tenet of Papert's constructionism (Papert, 1991) (which builds on constructivism). Thus, constructivist principles support the strategies of using more kinaesthetic and active approaches to teaching in the computer science classroom.

The “unplugged” style of activities which originated with the CS Unplugged project in New Zealand (Bell, Alexander, Freeman, & Grimley, 2009; Nishida et al., 2009) has resulted in many related, kinaesthetic activities which stimulate an understanding of a concept in a very concrete and practical way. CS4FN (Computer Science for Fun) (Curzon, McOwen, Cutts, & Bell, 2009) have generated many engaging activities and approaches by emphasising the importance of analogy as well as a kinaesthetic activity. Other research has highlighted the importance of providing a real world context for learning and relating it to students' interests and understanding and the value of a rich discourse regarding concepts (Grover & Pea, 2013).

Another key consideration in computer science pedagogy needs to be the development of computational thinking skills. Computational thinking was only recently popularised as a concept in 2006 by Wing (Wing, 2006) but teachers of computer science have been facilitating these skills in their students as long as this subject has been taught. For teachers in England, guidelines have been developed recently suggesting how computational thinking can be explicitly taught as part of the new curriculum (Curzon, Dorling, Ng, Selby, & Woollard, 2014).

Programming is the aspect of computer science in school which is perceived to be the most challenging. A range of activities can be used that allow students to collaborate and construct problem solutions. As an example, the following suggestions, drawing on a constructivist view of learning, are made by Van Gorp and Grissom:

- Code walkthroughs
- Writing algorithms in groups
- Insert comments in pairs into existing code
- Develop code from algorithm in pairs
- Find the bugs in code (Van Gorp & Grissom, 2001).

Reading and tracing code is also important in supporting the learning of programming has been demonstrated (Lopez, Whalley, Robbins, & Lister, 2008) and being able to do this is a pre-cursor to the problem-solving needed to write code (Lister et al., 2004). Lister later describes that novices need to be able to trace code with more than 50% accuracy before they can begin to confidently write programs of their own (Lister, 2011). In our study we were interested to see which types of strategies were being used in the classroom by the participants, and how they were supporting the development of strategies in reading, writing and tracing code.

This discussion about pedagogical approaches to teaching Computing can be related to the teachers’ pedagogical content knowledge (PCK), that is, the knowledge that a teachers has about how to teach their subject (Shulman, 1986).. But how does a teacher develop this PCK for teaching Computing? We hope initially through good initial teacher education, but also through professional development, sharing with other teachers, and learning from experience.

Research Focus

In this study we sought to ask a large number of active computing teachers how they recommend teaching the subject, in order to find out which strategies work well in practice. We balanced this with asking teachers about particular challenges they faced in teaching computing. Our research questions are quite simply the following:

- What pedagogical strategies do teachers report work well for teaching computer science in school?
- What challenges do teachers report that they face?

Black et al carried out a study in the UK where they asked teachers how they felt they could make the subject interesting (Black et al., 2013). The key aspects that they identified were the importance to teachers of making Computing fun and relevant. In carrying out our research we were interested to see whether the teachers' comments aligned with this study; in addition we asked more specifically for actual strategies that teachers use in their classroom that they feel to be effective.

This paper focuses purely on the teachers' perspective in addressing these questions. Diethelm et al emphasise the importance of the teachers' perspective to our understanding of computer science education as the teacher "*may work on many different abstraction levels or apply very different teaching methods for the same topic of the curriculum*" (Diethelm, Hubwieser, & Klaus, 2012). We wish to identify what these methods are, in particular identifying common themes that may help to provide guidance for teachers new to teaching the subject, as well as providing actual examples of teachers using effective strategies as we enter a phase of education when more and more students are studying computing in school.

In the next section the study carried out will be described. We will then report on the results of the content analysis that was used to analyse the responses of the teachers. Those aspects that require a whole new style of teaching for some teachers are identified. We then draw out how this can contribute to the general area of pedagogical content knowledge in our subject.

THE STUDY

The context: change in the curriculum

The UK has seen fast-paced change in the area of computer science education in the last few years (Brown et al., 2013; Brown, Sentance, Crick, & Humphreys, 2014). The state of computer science education is different in the four parts of the UK, with England having just implemented an ambitious new curriculum in Computing, to be taught from ages 5-16, and with a strong focus on computational thinking. This has been preceded by two years of preparation, as new qualifications were introduced and the draft curriculum proposed. Many schools and teachers in England had implemented elements of the Computing curriculum prior to the official starting date of the Computing Programme of Study of September 2014, as a void was left by the disapplication of ICT in January 2012 (Brown et al., 2014).

In the UK there is a strong subject association for computer science teachers, Computing At School (Brown et al., 2013). Through this grass-roots community of practice teachers are able to share resources, share experiences and attend local events. The participants of this study were to a very large extent members of this community. In the data collected in this study, they describe the experiences, successful strategies, and also the frustrations, of teachers who have begun to teach Computing in school over the last few years.

The Computing Programme of Study for the new English Curriculum (Department for Education, 2013) is based on computational thinking principles, and thus teachers of computer science welcome guidance on how to deliver computational thinking skills; which is beginning to emerge (Curzon et al., 2014).

Survey of teachers' perspectives

A wide-ranging survey was carried out of members of Computing At School. As one part of this, teachers were asked if they optionally wished to contribute free-text answers to the following four questions about their teaching.

1. What good techniques/strategies have you found for helping students to understand programming?
2. Please describe any good techniques/strategies you use for helping students to understand other aspects of Computing?
3. What difficulties, if any, have you experienced teaching programming?
4. What difficulties, if any, have you experienced teaching other aspects of Computing?

In the context of this survey, teachers in the England understand "other aspects of Computing" to be non-programming topics in the curriculum, which include learning about hardware, networking, data representation and logic (Department for Education, 2013) .

The survey was publicised via the Computing at School forum, as well as through social media channels. 1417 members completed the wider survey, with 357 teachers contributing at least one free text answer to the free text questions. In this paper we focus only on the 357 responses given by this self-selecting group of teachers but include reference to their other answers to survey questions where relevant.

The data was collected by an online questionnaire which was then input into qualitative data analysis software. The data consisted of the four free text questions described above, plus responses that these teachers gave to the other questions in the wider survey.

Study Participants

The 357 teachers responding to the questions were from different phases of education, including primary (ages 4-11), secondary (ages 11-18) and post-secondary education (ages 16 to 18). The majority (76%) of the teachers were from secondary education, with 20% from primary and middle schools and 4% from post-secondary education.

Teachers were asked to rate their confidence in being able to deliver the new Computing curriculum on a Likert scale from 0 to 10. This self-selecting group were largely confident in their Computing teaching, with 85% rating their confidence at 6 or more out of 10. Their confidence overall was greater than the confidence levels of the wider population completing the larger survey. The general confidence in the teaching of Computing will have contributed to their willingness to participate in a free text questionnaire on their practice and also will have a bearing on the content on their responses. This indicates that they may not be `typical' of the whole teacher population, but represent teachers who are more comfortable teaching Computing.

ANALYSIS

The data was initially coded in an inductive manner with respect to emerging themes, following the guidelines in (Mayring, 2000). The themes were then grouped to facilitate further analysis. The data was re-coded and verified by two researchers to ensure agreement on the interpretation of the teachers' statements. The coded answers were then examined in relation to the level of confidence that the participants had in the context of teaching computing.

Table 1 provides an overview of the particular strategies that teachers mentioned when describing their teaching.

Table 1: Strategies used by teachers teaching programming and non-programming aspects of the curriculum

Coded strategy (programming)	**% mentions**
Practice/little & often	14%
Unplugged/ teach away from computer	13%
Use of particular software	13%
Scaffolding/modifying code	12%
Varied activities	11%
Exercises around coding	10%
Use lots of examples	9%
Relate activities to the real world	8%
Demonstration & modelling	8%
Peer mentoring	8%
Learn through exploring	7%

Coded strategy (non-programming)	**% mentions**
Unplugged/ teach away from computer	19%
Hands-on activities	17%
Relate activities to real world	10%
Show videos	9%
Work in groups	8%
Use published resources	6%

This table shows that teachers emphasised unplugged, hands-on, contextualised activities and the importance of lots of practice. Approximately the same number of teachers mentioned working on tasks away from the computer as mentioned a particular software package that they used. In addition, a high percentage of teachers (13%) referred to particular software that helped them to teach programming and other concepts. The study looks entirely at free text comments with suggestion within the question; there are themes emerging quite clearly from this data around using activities away from the computer that promote understanding. These will be discussed in more depth in the next section.

Teachers reported a range of different challenges that they faced when teaching Computing:

- Students not understanding / having difficulty

- Teachers’ own subject knowledge
- Challenge of differentiation
- Lack of resources
- Technical problems
- Students’ abilities in problem-solving tasks

Some of the challenges mentioned relate to the teachers’ own difficulties – for example, not being confident in the subject matter or not being able to differentiate sufficiently for a mixed-ability group, and other comments focus on the fact that the students have difficulty understanding the material and in problem solving. The data showed overall three areas of challenge for teachers: their confidence in teaching computing as a subject, the difficulties (or perceived difficulties) inherent in the subject matter and the issue of having sufficient resources, including technical support, in the classroom.

The remainder of the paper focuses particularly on the good strategies that teachers report that work well for them.

KEY THEMES EMERGING

In this section we address specific themes that emerged from strategies used by teachers. Most of the individual strategies suggested by teachers could be grouped into a series of five themes, which are (in no particular order):

- Contextualisation of learning
- Collaborative working
- Computational thinking
- Code tracing and scaffolding
- Learning away from the computer

Typical quotes from teachers illustrate these themes.

Contextualisation of learning

Teachers talk about relating computing content to other aspects of the curriculum; they give examples of both relating what is being learned in computing to other subjects taught at school and also to concepts from home (so relating to real-life). The quote below is a typical example:

> “*Scale it back to basics and use real-life examples for the activities e.g. making tea. Use lots of visual aids to help pupils and online resources to help scaffold activities*.” (case 233, secondary teacher, confidence 7).

It is interesting to examine the range of ways in which teachers talk about the contextualisation of learning.

Collaborative Working

The analysis of teachers’ qualitative responses highlights a variety of collaborative working strategies that they use within the classroom and would promote to other computer science teachers. These collaborative strategies included: team work, peer mentor, paired programming and collaboration. These strategies resonate with the concept of computational participation (Kafai & Burke, 2014) and strategies proposed to develop this within the classroom. In addition individual teachers commented on the positive motivational impact that collaborative working has on individuals, small groups and the class itself.

> "…*Developing digital leaders in students who can support others.* …" (case 345, primary teacher, confidence 9).

> "*Decomposing sample problems together as a class then team-coding …they can use peers for discussion of specific problems.* …" (case 12, secondary teacher, confidence 9).

Computational Thinking

Analysis of teachers' qualitative responses indicates a number of computational thinking concepts and processes that teachers want to promote and develop their students' competence in through using a variety of teaching and learning activities. These concepts and processes include: logic (algorithmic) thinking, decomposition, problem solving and abstraction (Brennan & Resnick, 2012; Curzon et al., 2014).

> "*Breaking down the problem then breaking it down again then breaking it down again...* …" (case 109, secondary teacher, confidence 8).

> "*Organise the learning so that the pupils develop their programming skills using decomposition and abstraction. …."* (case 265, secondary teacher, confidence 10).

Code tracing and scaffolding

Closely related to the theme of computational thinking are the strategies that teachers use to help their students understand program code. One teacher describes a range of types of strategies used to support students learning programming, that involve:

> "… *giving code on paper not electronically, so they have to type it in, think about what they are typing and fix the errors that occur when trying to compile the program …* (case 113, secondary teacher, confidence 7).

> "*Discussion of what a specific algorithm does, then running trace tables on small programs …"* (case 310, secondary teacher, confidence 7).

Other strategies described included "scaffolding" as the student is given part of a program to extend, and programs to debug. Typing in code to give more chance that the program would work, but involving debugging errors caused by transcription errors is another supportive strategy for early programmers reported by teachers.

Unplugged-style or kinaesthetic activities

A significant proportion of teachers mentioned, unprompted, that they try to support students' understanding by using physical, or unplugged-style activities in the classroom. One teacher gives two examples of teaching different topics using physical visual-aids to support the learning:

> "*For example I use clear plastic drinking cups as memory locations and label them as variables or when demonstrating an algorithm like bubble sort add data (on pieces of paper)."* (case 229, secondary teacher, confidence 9)

Many of these activities are designed to promote both collaboration and computational thinking skills. In fact, whether the activity takes place on the computer or not may not be what is interesting. The key link between the statements made by teachers seemed to be their impression that actually physically being engaged in the activity was conducive to the students' learning. This is an area which needs further research to establish.

Putting it all together

As reported earlier, Diethelm (2012) discusses the way that teachers use a variety of strategies as part of their pedagogical content knowledge for the subject. This was most definitely the case with the data that we looked at. A number of teachers described strategies that they used for teaching which included a mixture of the types of approaches already described – these have been underlined in this comment:

> *"… Provide some examples which have errors to be fixed - or examples that need re-writing … more efficient perhaps and get pupils to explain their decisions) …Get them working away from the computer at times to ensure they consider the steps of the program they are undertaking rather than just hacking away on the computer . Ask questions and get them to explain program concepts i.e. to vocalise an understanding … Discussion is important at times - not just doing*." (case 132, secondary teacher, confidence 10).

Here it can be seen that the teacher (who self-reports as having a high level of confidence in teaching Computing) is combining strategies around code exercises, using discussion (collaboration and computational thinking), and working away from the computer. The key for this teacher seems to be to utilise a variety of teaching strategies to support learning, rather than relying on one particular strategy. What is key in this description is the need for students to reflect on what they are learning in computing and be able to articulate it.

DISCUSSION

Examining the statements of teachers as they report what strategies work well for them in teaching Computing has enabled us to draw out particular themes. Ben-Ari (1998) advised teachers: "*Don't run to the computer*", and it seems that teachers are taking this advice in using a variety of other strategies to get concepts across. In addition, the use of collaborative work, peer mentoring, pair programming and other strategies is helping teachers to establish computational thinking skills in young students.

The teachers participating in the survey are, in the majority, members of Computing At School and as such have access to a lively and supportive grass-roots community of teachers with whom they can exchange ideas and classroom resources. It could be tentatively suggested that the presence and nature of this community of practice may also have an impact on the commonality between the approaches teachers are successfully using, but this cannot be verified from the data analysed.

It cannot be assumed that the teaching approaches described by teachers in this study are representative of all teachers teaching the English Computing curriculum. The participants are self-selecting and have mostly reported themselves as being confident in their delivery of Computing so the data gives us reports of good practice. We are also not able to provide evidence for which of these suggested approaches is more effective in helping students to learn without more empirical research; thus another useful angle on this question would be to examine students' own perspectives on how Computing is taught.

CONCLUSION

In this paper, we have identified a range of pedagogical strategies that teachers use in practice which can be categorised into the five areas of contextualised learning, computational thinking skills development, code manipulation, working collaboratively and learning away from the computer.

The study exemplifies a link between theory and practice as teachers report that they use strategies for teaching Computing that previous research has suggested to be of value. In addition, teachers who self-report as "confident" use a combination of these strategies to support their students' understanding.

We suggest that focusing on the use of a range of these strategies could help teachers to feel more confident in the computer science classroom. We believe the results of this study are significant in that they will give novice teachers (or experienced teachers new to computing) some ideas about pedagogical approaches that work. More research will be needed as to what extent students perceive these strategies to be effective also.

REFERENCES

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology, 13*(1)

Ben-Ari, M. (1998). *Constructivism in computer science education*. Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education. Atlanta, Georgia, United States: ACM.

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P. W., & Meagher, L. R. (2013). Making computing interesting to school students. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '13).* Canterbury, UK. 255. doi:10.1145/2462476.2466519

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. in . *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada.,* Vancouver, Canada.

Brown, N., Kölling, M., Crick, T., Jones, S. P., Humphreys, S., & Sentance, S. (2013). Bringing computer science back into schools: Lessons from the UK. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013),* 269-274.

Brown, N., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions of Computing Education, 14*(2)

Bruner, J. (1996). *Towards a theory of instruction*. Cambridge, MA: Harvard University Press.

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: A framework.* Available at: http://community.computingatschool.org.uk/files/3517/original.pdf).Unpublished report.

Curzon, P., McOwen, P., Cutts, Q., & Bell, T. (2009). Enthusing and inspiring with reusable kinaesthetic activities. *Proceedings of the ITICSE 2009,*

Department for Education. (2013). *National curriculum for England: Computing programme of study.*. London, England: Department for Education.

Dewey, J. (1938). *Experiential education*. New York: Collier Books.

Diethelm, I., Hubwieser, P., & Klaus, R. (2012). *Students, teachers and phenomena: Educational reconstruction for computer science education*. Koli, Finland: ACM.

Grover, S., & Pea, R. (2013). *Using a discourse-intensive pedagogy and Android's App Inventor for introducing computational concepts to middle school students*. Proceedings of the 44th SIGCSE technical symposium on Computer science education. ACM.

Kafai, Y. B., & Burke, Q. (2014). *Connected code why children need to learn programming.* Cambridge MA: MIT Press.

Lister, R. (2011). Concrete and other neo-piagetian forms of reasoning in the novice programmer. *Proceedings of the Thirteenth Australasian Computing Education Conference - Volume 114,* Perth, Australia. 9-18.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education,* Leeds, United Kingdom. 119-150.

Lopez, M., Whalley, J., Robbins, P., & Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. *Proceedings of the Fourth International Workshop on Computing Education Research,* Sydney, Australia. 101-112. doi:10.1145/1404520.1404531

Mayring, P. (2000). Qualitative content analysis. *Forum of Qualitative Social Research, 1*(2)

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09),* 231. doi:10.1145/1508865.1508951

Papert, S. (1991). *Constructionism* Ablex Publishing.

Piaget, J. (1950). *The psychology of intelligence*. Cambridge, MA: Harvard University Press.

Sentance, S., Dorling, M., & McNicol, A. (2013). Computer science in secondary schools in the UK: Ways to empower teachers. In I. Diethelm, & R. Mittermeir (Eds.), *Informatics in schools: Sustainable informatics education for pupils of all ages.Lecture notes in computer science* (pp. 15-30) Springer-Verlag.

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *American Educational Review, 15*(2)

Thompson, D., & Bell, T. (2013). *Adoption of new computer science high school standards by New Zealand teachers* ACM.

Van Gorp, M. J., & Grissom, S. (2001). An empirical evaluation of using constructive classroom activities to teach introductory programming. *Computer Science Education, 11*(3), 247-260.

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