No images? Click here

SAPIENTIA

From ICT for Education

Everything you need to know about Computing, the curriculum, and the classroom

Welcome to Sapientia, ICT for Education's termly newsletter that provides education professionals with thought leadership, an insight into hot topics, and practical guidance on how to implement new technologies and techniques to improve teaching and learning.

This edition of Sapienta leads with an article by **Jill Hodges, Head of Amazon Future Engineer UK**, who discusses the need for the education system to catch up with the pace of technological change to ensure students can fulfil and enjoy careers of the future. In support, she notes initiatives such as Amazon Future Engineer, a programme designed to give students experience of emerging tech fields and teachers the resources to help the students develop in-demand skills.

An article by **Miles Berry, Professor of Computing Education at the University of Roehampton**, considers the potential of adaptive, or agile, teaching that accounts for varying levels of prior knowledge based on detailed assessment data. This data, says Berry, is more informative for teaching than 'flight paths' or target grades, with target grades perhaps being harmful as a low target could lead to a low expectation of what a pupil can achieve, and thus a lack of challenge.

He suggests questions in class offer a powerful way to adapt teaching, as well as inform teachers about what pupils already know and can do. With the right atmosphere in class, where critical thinking and curiosity are valued, and where pupils share the responsibility for learning, questions can be a powerful tool for learning.

In this edition of Sapientia, we also hear from **Paul Curzon, Professor of Computer Science in the School of Electronic Engineering and Computer Science at Queen Mary University of London**. Paul writes about improving lesson plans with sketchy semantic profiles. These use semantic waves that are based on the idea that a good explanation follows a wave structure, moving between a technical concept that is hard to understand to concrete things the learner already understands and, critically, back again.

What matters, he says and illustrates in the article, is that a lesson involves both unpacking concepts into things that are easy for students to understand and repacking them into technical terminology. As well as developing students' skills, this allows teachers to reflect on and improve lesson plans quickly and easily.

Meet and hear Jill Hodges and Miles Berry at the ICTfE Manchester University seminar on March 19th Meet and hear Paul Curzon at the QMUL seminar on April 23rd

To keep pace with the changes, challenges and opportunities in the primary and secondary education sectors, register for ICT for Education's termly newsletter here or e.mail il@ictforeducation.co.uk.

Sarah Underwood, Editor - ICT for Education

Preparing the Next Generation for Careers

of the Future

By Jill Hodges, Head of Amazon Future Engineer UK.

We live in a world where technology is evolving at an exponential pace. Fields like artificial intelligence, robotics, and genomics are transforming industries right before our eyes. Research that Amazon commissioned with Capital Economics suggests jobs that require AI and Machine Learning skills are expected to increase by 40% between 2022 and 2027 and could contribute £71 billion a year to the UK economy. [Source]

Yet our education system moves much slower than the pace of technological change. While almost half of students express an interest in careers using computer science and AI, 68% of secondary school teachers say that the children in their classes don't have enough information to understand those careers. If students lack exposure to future-focused careers, how can we expect them to envision themselves in those roles?

Closing the Awareness Gap

Education must evolve to close this awareness gap. Students need exposure to technology careers long before they reach university or the job market. It's never too early to start developing students' problem-solving skills, and to give them opportunities to develop exposure and confidence so they see that they can 'belong' in the careers of the future.

Initiatives like Amazon Future Engineer give students exposure into emerging tech fields through hands-on computer science courses, coding challenges, and virtual school trips. Students learn in-demand skills while seeing first-hand how technology can be an avenue for creativity, communication, and social impact. Exposure of this kind gets students excited about using tech for good – not just as passive consumers of devices, but as empowered creators and architects of future technologies.

The earlier we foster passion and talent in science, maths and technology, the more underrepresented students may see these subjects as springboards for future careers versus obstacles to avoid. Closing the awareness gap will help dispel notions that tech careers are only meant for certain types of students.

Democratising Access

However, awareness alone is insufficient if the right opportunities are not available to underserved schools and communities. We need coordinated efforts across education, government, philanthropy and business to ensure high-quality STEM education reaches every student.

Amazon Future Engineer is one programme committed to this mission. Teacher-led training and inspiring curricula help educators develop their students. Young people from primary through to secondary school can take virtual career tours or hear directly from Amazon employees about their job roles and how they got there. Women who want to study Computer Science and related courses at university are eligible for bursaries to help eliminate financial hurdles. Initiatives focused on access and inclusion will help build a diverse talent pool ready to lead technological innovation.

Preparing the Next Generation

The jobs fueling UK prosperity should reflect the diversity and creativity of our youth. With greater career awareness and technological access, we empower students from all backgrounds to envision themselves as tomorrow's innovators. Our nation's economic future depends on readying young people to lead in the digital economy and on the equality of access to opportunity in the sector. Preparing the next generation with technology fluency and career exposure is an investment in their limitless potential and our shared future.

Register to meet and hear Jill at the [ICTfE Manchester University seminar on March 19th]

Jill Hodges leads Amazon UK's educational outreach programme, Amazon Future Engineer. Since launch in 2019 AFE has engaged with over 400K students and their teachers, with engaging opportunities to gain skills and confidence to engage with Careers of the Future.

Amazon Future Engineer focuses on working with educators to prepare students, especially those from underserved and underrepresented communities, for careers of the future. We provide teacher training, classroom resources and interactive career exploration opportunities to primary and secondary school teachers and their students. You can sign up here for updates on Amazon Future Engineer

How to adopt adaptive teaching

By Professor Miles Berry, Professor of Computing Education at the University of Roehampton.

Once upon a time, software was developed using a 'waterfall' method, where the requirements were gathered and the software was designed, implemented, tested and deployed. This approach is still taught in GCSE and A Level computer science classes, and curriculum development often follows a similar approach. Software engineering has moved on, recognising the value of an iterative approach, also mirrored in teaching, where the results of implementation are analysed to inform the next cycle of the design.

The best software development is now done using agile methods, which recognise that while processes, tools, documentation, agreements and plans are important, they're less important than individuals, interactions, getting things working, collaborating and responding to change. In school, perhaps we should place less importance on delivering content, and more on adapting teaching and curriculum to our pupils and what they know already, while ensuring that all reach high standards. Adaptive teaching is our equivalent of agile software development.

There are any number of factors that contribute to a pupil's success in computing, but there's no evidence to suggest there's a coding gene. Some pupils do seem to learn faster than others, and some start a class with more prior knowledge, but if we explain things well and provide meaningful activities and exercises, and if pupils pay attention and work hard, they can all learn.

Until recently, there was a focus on differentiation, which often involved teachers providing different activities based on perceived ability. This involved a lot of work for teachers, and it didn't really help overall. Although it produced a bit more progress for those in top groups, this was at the detriment of those in lower groups, and this was particularly the case for those from disadvantaged backgrounds.

The emphasis now is on adaptive, or agile, teaching, and increasing recognition that ability or prior attainment is just one factor to consider in better meeting the needs of learners. Prior knowledge does make a difference, as new knowledge is built on existing knowledge, and the more a pupil knows, the easier it is for them to learn more. Good adaptive teaching must account for varying levels of prior knowledge, for which detailed assessment data is needed. This sort of granular assessment, perhaps which pupils have mastered variables in Scratch, is far more informative for teaching than 'flight paths' or target grades. Target grades might even be harmful, as a low target might lead to a low expectation of what a pupil can achieve, and thus a lack of challenge.

There are differences between experts and novices. Most pupils are novices when they're learning computing. While clear explanations, direct instruction, worked examples and lots of practice are effective for novices, they can be counterproductive for experts. For experts, experimentation, exploration and discovery are more productive, so difficult problems and independent study might well be part of learning for these pupils – there are excellent online study materials, coding challenges and online communities for well-motivated learners.

Some novices grasp things more slowly than others, and thus might benefit from extra support or scaffolding. The ‘mastery’ approach now common in maths teaching, with the class moving on when they’re ready but with tailored support for those who need it, can work well in computing too.

Pupils with special educational needs or disabilities often need more support, and sometimes teachers do need to adapt the way things are taught, or the way pupils might respond, to take these needs into account. Technology, such as speech synthesis and recognition, and large language models like GPT, can help here, particularly when pupils use these tools themselves. Some special needs, such as autism, are no barrier to doing well in computing. Pupils learning English as an additional language face additional cognitive load, and technology, in the form of Google Translate and similar tools, can help – and there’s nothing wrong with pupils swapping the language pack over in Scratch to program in their mother tongue.

Socio-economic factors also have an impact. Pupils without access to a computer or the internet at home won’t have had as much opportunity to develop tech skills as their peers – for those receiving it, pupil premium funding could help.

Although there’s limited evidence of its impact on learning outcomes, the insights of Universal Design for Learning (UDL) are helpful, so provide multiple means of representation, action, expression and engagement. If what matters is that pupils can apply the programming construct of selection, does it matter if they use Scratch or Python, make a game or a quiz program, a written explanation or narrated screen recording? Chat GPT and similar are effective at taking materials in one form and adapting them to another.

Questions in class offer a powerful way to adapt teaching, as well as inform us about what pupils already know and can do. With the right atmosphere in class, where critical thinking and curiosity are valued, and where pupils share the responsibility for learning, questions can be a powerful tool for learning. You can pitch your questions at the sweet spot to take pupils forward, linking what they know already to what you’re trying to teach, eliciting their current understanding, getting them to think more critically and speak more precisely, and helping them make connections between different ideas.

For those who are struggling, recall and retrieval questions can help to build confidence, and for those who've already 'got it', open-ended, fertile questions can do much to extend their thinking and understanding. The key thing is to increase what Doug Lemov calls 'ratio', the amount of time pupils are spending thinking and talking about the subject and decrease the amount of time the teacher is doing the thinking for them.

Don't let adaptive teaching diminish the challenge of learning computing, encourage all pupils to rise to the challenge. Compare coding with gaming: it's the challenge that makes it fun.

Register to meet and hear Miles at **the ICTfE Manchester University seminar on March 19th**

Professor Miles Berry

Miles Berry is Professor of Computing Education at the University of Roehampton. Before joining Roehampton, he spent 18 years in schools, including a period as a head teacher. He has contributed to a wide range of computing projects, including the computing programmes of study in the National Curriculum, Barefoot Computing and Switched On Computing. He serves on the boards of Computing At School, the BCS Academy of Computing, and the National Centre for Computing Education, and is a regular keynote speaker and international consultant on curriculum and professional development. He is @mberry on Twitter and find out more on milesberry.net

Improving fun lesson plans with sketchy profiling

By Paul Curzon, Professor of Computer Science in the School of Electronic Engineering and Computer Science at Queen Mary University of London

Games, magic and puzzles provide a fun way for students to learn about concepts, but how do we do this effectively to ensure it is more than play? One answer is semantic waves, which are based on the idea that a good explanation follows a wave structure, moving between a technical concept that is hard to understand to concrete things the learner already understands and, critically, back again (Figure 1). What matters is that a lesson involves both unpacking the concepts into things that are easy to understand and repacking them again into the technical terminology.

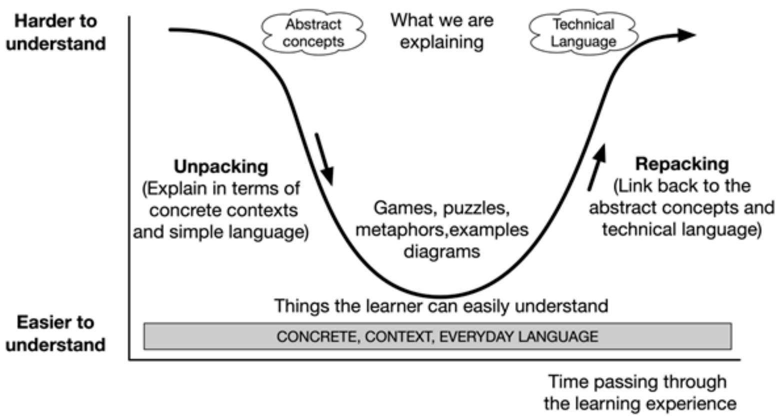


Figure 1: *A semantic profile of an activity in the form of a wave.*

The profile of a less effective lesson might have all the abstract and technical steps but it flatlines high. The lesson never links to everyday things the learners already understand. A lesson can also flatline low. It remains throughout about a puzzle, say, and never really links to the technical concept, so no one learns about it. The lesson is just play. More subtle is the down escalator where the abstract concept is unpacked using a concrete example, or into easy to understand ideas, but the activity ends there. There is no repacking to link the concrete back to the concept actually being learned. Again, the learners leave thinking it was just play.

Algorithms are a core concept that primary school students are expected to understand. You could give students a word search puzzle as a way to learn about algorithms, as following an algorithm is a way to solve such a puzzle. A student might scan rows for the first letter of a word in the word list then scan in each direction for the rest of the word. If the students are just given a word search to do, though, they may learn nothing. They are being flatlined low.

If, instead, we outline the aim of the lesson as being about algorithms, ask if any students know what an algorithm is, explain that it is a series of steps that guarantees a result, point out that it can be used to solve puzzles, and then set a word search to do, we have descended a wave from top to bottom in a series of steps. However, if we stop there, we have a down escalator and students may learn little about algorithms.

If, as they are doing the word search, we prompt them to think about how they are doing it – are they randomly scanning, hoping words jump out or are they following an organised method that guarantees finding a word? – we are now making them move up a semantic wave, making links between a word search and the idea of an algorithm. Once some of the students' ideas have been shared, we can suggest they use a way that does guarantee finding a word and even write down the steps they are following. We are then repacking a little more. Then, as they solve the puzzle, they repack further. Finally, we ask them to summarise what an algorithm is and how it can be used to solve a word search before we wrap up with a final summary.

This considers algorithms in the context of puzzles. Ultimately, students need to connect the concept to programs. A final point could be to make this link with a further activity that builds on the idea and creates a new wave based on a technical programming example that links word search algorithms to writing a program, perhaps a program to find a given word in a list.

We need to take care that the steps focus on the thing we want the students to be unpacking or repacking, here, algorithms / steps that guarantee a result. With a word search, that may mean we do not want them struggling with spelling instead of thinking of ways to guarantee finding a word. We should ensure words used are at the right level for each student.

We can build a way to improve lesson plans from these ideas using sketchy semantic profiling. This also allows us to quickly and easily reflect on and improve lesson plans. It involves drawing a rough profile based on the main steps of a lesson plan and then asking some simple questions. Draw a curve downwards if the next step uses more concrete examples or more everyday language or context. Draw a curve upwards if the next step becomes more abstract or more technical. Figure 2 shows a semantic profile of a simple word search lesson.

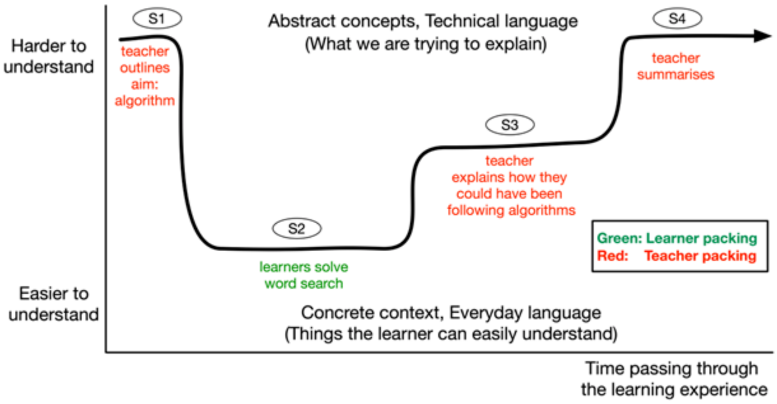


Figure 2: *A sketchy semantic profile of a simple word search lesson plan.*

The questions are:

1. Does the profile drawn have a rough 'u' or 'n' shape?

If we are flatlining, add steps to make links that step up or down.

If following a down escalator, add steps that link back up to the concept.

If the curve drops off a cliff edge, going from aim to activity, add intermediary steps.

2. How high and how low does the curve go?

If we don't link all the way back to the abstract concepts add a final step to do so.

If the examples are very technical, link to some everyday context.

3. Who is doing the unpacking and repacking: teacher or learner?

If the packing is mostly done by the teacher, make the learner do some steps.

4. Does each step focus on the target learning outcome of the lesson?

If a step directs students to think about something else that is not our immediate aim, modify the activity so it meets the aim.

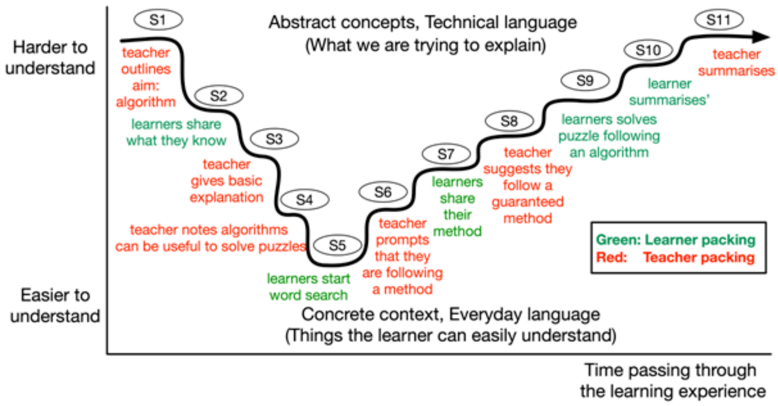


Figure 3: *A sketchy semantic profile of an improved word search lesson plan avoiding the cliff edge and with more student repacking.*

Draw sketchy semantic profiles of your lesson plans, then think about these questions. You could make your lessons both more effective and much more fun.

For more on Computer Science for fun, visit: [cs4fn.blog]

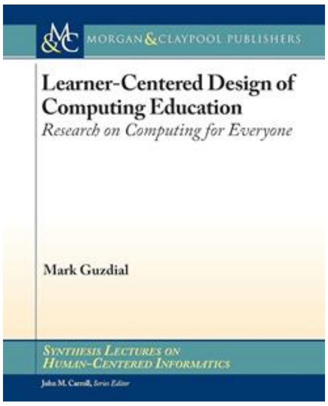
For fun classroom activities, visit: [teachinglondoncomputing.org]

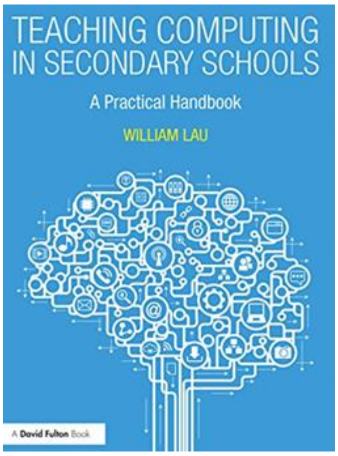
Register to meet and hear Paul at the [ICTfE QMUL seminar on April 23rd]

Paul Curzon

Paul Curzon is Professor of Computer Science in the School of Electronic Engineering and Computer Science at Queen Mary University of London.

Find out more about Professor Paul Curzon by visiting: https://www.eecs.qmul.ac.uk/~pc/

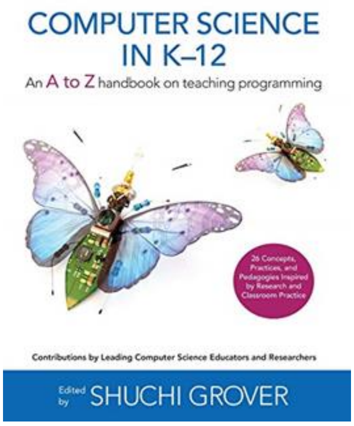


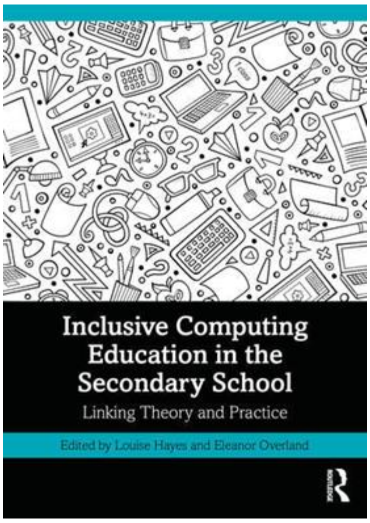


Recommended reading

Miles Berry, Professor of Computing Education at the University of Roehampton recommends Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. Morgan & Claypool Publishers. Mark Guzdial's accessible and authoritative view of the different reasons why people may want to learn computing, and the implications of these for what and how we teach them.

Miles also recommends Lau, W. (2017). Teaching Computing in Secondary Schools: A Practical Handbook. Routledge. Secondary computing teacher William Lau gives an excellent summary of how computing can be taught in alignment with what we know about learning, with lots of practical suggestions for how we can adapt teaching to meet learners' different needs.





Paul Curzon, Professor of Computer Science in the School of Electronic Engineering and Computer Science at Queen Mary University of London recommends An A to Z handbook on teaching programming. Ed. Grover, S. Edfinity.

A book providing practical advice on the pedagogy of how to teach programming at school level based on research by a wide range of experts.

Paul also recommends Computer Science Education: Perspectives on Teaching and Learning in School, edited by Sue Sentance, Erik Barendsen, Carsten Schulte, 2nd Edition. Bloomsbury Academic.

An overview of wide ranging aspects of how to teach computer science, covering topics from pedagogy and assessment to equity and inclusion.

We also recommend Inclusive Computing Education in the Secondary School. Edited by **Louise Hayes**, Senior Lecturer in Initial Teacher Education at Manchester Metropolitan University and Eleanor Overland, Director of Quality Assurance for Initial Teacher Education at Manchester Metropolitan University.

SAVE 20% with code

SMA40

Learn, Share, Enjoy

ICT for Education can help you get the most out of technology in your classroom and in your school. Our free to attend conferences and seminars provide relevant, innovative, informative content delivered by experienced, knowledgeable, respected speakers able to relate to and understand the challenges faced by those responsible for giving learners the best opportunities in life.

Attend a Conference

Norwich - 10/05/2024

Brighton - 14/06/2024

Newcastle - 20/09/2024

Salford - 10/10/2024

Birmingham - 08/11/2024

Reserve your conference place

Carrow Road Stadium, Norwich

Amex Stadium, Brighton

Attend a Seminar

Manchester - 19/03/24

Queen Mary - 23/04/24

Manchester - 21/05/24

Aston - 25/06/24

Book your seminar place now

Manchester University

Queen Mary University, London

SAPIENTIA

From ICT for Education

Forward