The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3290404

<http://cacm.acm.org/blogs/blog-cacm>

Answering Children's Questions About Computers

Judy Robertson addresses the disconnect between what children are taught about computers and what they actually wish to know.

Judy Robertson
What Children
Want to Know
About Computers

[https://cacm.acm.org/blogs/blog-cacm/231993-](https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext)
[what-children-want-to-know-about-](https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext)
[computers/fulltext](https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext)

[what-children-want-to-know-about-](https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext)
[computers/fulltext](https://cacm.acm.org/blogs/blog-cacm/231993-what-children-want-to-know-about-computers/fulltext)

October 19, 2018

There is a mismatch between what we teach children about computing at school and what they want to know.

More than a decade ago, computer science educators coined the phrase *computational thinking* to refer to the unique cleverness of the way computer scientists approach problem solving. "Our thinking is based on abstraction, decomposition, generalization, and pattern matching," we said, "and everyone will find it useful to think like this in their everyday lives. So please stop asking us to fix your printer." Computational thinking has been a hugely successful idea and is now taught at school in many countries across the world.

Although I welcome the positioning of computer science as a respectable, influential intellectual discipline, in

my view computational thinking has abstracted us too far away from the heart of computation—the machine. The world would be a tedious place if

we had to do all our computational thinking ourselves; that is why we invented computers in the first place. Yet, the new school curricula across the world have lost focus on hardware and how code executes on it.

When visiting a series of eight primary school classrooms recently, I talked to children (5 to 12 years old) about how computers work. They drew pictures of what they thought is inside a computer, and then we discussed the drawings as a class.

Many of the children knew the names of the components within a computer: a chip, memory, a disc, and

Drawing 1.

Drawing 2.

they were often insistent that there should be a fan in there. They knew that there would be wires inside, and that it would need a battery to make it work. The child who created Drawing 1 has made a nice job of piecing together a possible design from what they knew about computers—can you spot what is missing, though?

The artist of Drawing 2 knows there is a chip inside (made by HP, in this case) and to their credit, they know there is code, too. Notice that the code is not physically located on the memory or the chip, but somewhere in the wires. In general, there was some puzzlement about how code related to the computer, as exemplified by the artist of Drawing 3, who confessed, “I know a computer is full of code and all devices. I am not sure what it looked like, so I just scribbled.”

Often, the children spent a while thinking about what is outside the computer and how information might get inside. It was quite common to see pictures in which the artist had folded the page to show this distinction but it was often a mystery how pressing a key or touching the screen might make something happen in the computer. Children who had spent time tinkering with computers at home had an advantage here: “I broke my keyboard once and I saw what was inside. It would send a signal from key to computer to the monitor.”

What the pictures and subsequent classroom discussions told me is that the children know names of components within a computer, and possibly

Drawing 3.

some isolated facts about them. None of the pictures showed accurately how the components work together to perform computation, although the children were ready and willing to reason about this with their classmates. Although some of the children had programmed in the visual programming language, none of them knew how the commands they wrote in Scratch would be executed in the hardware inside a computer. One boy, who had been learning about variables in Scratch the previous day, wanted to know whether if he looked in his computer, he would really see apps with boxes full of variables in them. I love that question, because it reveals the mysterious boundary between intangible, invisible information and the small lump of silicon that processes it.

To be clear, I am not criticizing the children, who were curious, interested, and made perfectly reasonable inferences based on the facts they picked up in their everyday lives. But I think that computer science educators can do better here. Our discipline is built upon the remarkable fact that we can write instructions in a representation that makes sense to humans, and then automatically translate them into an equivalent representation that can be followed by a machine dumbly switching electrical pulses on and off. Children are not going to be able to figure that out for themselves by dissecting old computers or by making the Scratch cat dance. We need to get better at explicitly explaining this in interesting ways.

Children are currently piecing to-

gether their everyday experiences with technology with facts that adults tell them to try to make sense of how computers work. This can lead to some confusion, particularly if the adults in their lives are also unsure. One child thought, for example, that if you paid more money, then it would make Wi-Fi stronger. Others were curious about how Wi-Fi works on a train, and whether you really need to stop using your phone on a plane. A student advised the class that if we needed to save space on our phones, then we should delete videos from YouTube. The children, like most Windows users, wanted to know why their computers “freeze,” speculating that it could be because the chip is asleep or that too many people are using Wi-Fi. There was also a sense of wonderment and curiosity. A young boy was fascinated when he read about supercomputers and wanted to know more: Do supercomputers have really big chips in them? A class of 11-year-olds gravely debated whether people would be more or less clever if the computer had never been invented. These are the sorts of questions about computers that children want to explore. It's our job as computer scientists, and as educators, to help them.

[This article was based on a keynote talk at the Workshop in Primary and Secondary Computing Education (WiPSCE) 2018.]

Judy Robertson is professor of Digital Learning at the University of Edinburgh, U.K.

© 2019 ACM 0001-0782/19/1 \$15.00