DOI:10.1145/3265747

Mike Tissenbaum, Josh Sheldon, and Hal Abelson

Viewpoint
From Computational Thinking to Computational Action

Envisioning computing education that both teaches and empowers.

COMPUTATIONAL ACTION, A new framing for computing education, proposes that while learning about computing, young people should also have opportunities to create with computing that have direct impact on their lives and their communities. In this Viewpoint, we outline two key dimensions of computational action—computational identity and digital empowerment—and further argue that by focusing on computational action in addition to computational thinking, we can make computing education more inclusive, motivating, and empowering for young learners. Learners have the capacity to develop computational products that can have authentic impact in their lives from the moment they begin learning to code, all they need is to be situated in contexts that allow them to do so.

Too often, K–12 computing education has been driven by an emphasis on kids learning the "fundamentals" of programming. Even more progressive CS education that centers around the development of learners' computational thinking has largely focused on learners understanding the nuanced elements of computation, such as variables, loops, conditionals, parallelism, operators, and data handling.[10] This initial focus on the concepts and processes of computing, leaving real-world applications for "later" runs the risk of making learners feel that computing is not important for them to learn. It begs the question far too many math or physics students have asked, "When will we use this in our lives?"[1]

While there have been attempts to situate computing education in real-world contexts and problems, they are often generic (for example, designing checkout systems for supermarkets) and fail to connect to the specific personal interests and lives of learners. Though real-world application of their work is valuable for all learners, not providing opportunities to develop computational solutions with real-world potential is particularly problematic for young women and youth from nondominant groups. For these groups, who have been traditionally underrepresented in the computing fields, it has been observed that a sense of fitting into and belonging to the broader computing community is closely tied to being able to develop computational solutions that matter to themselves and those in their communities.[8] By connecting with students' real lives, we can help them develop a critical consciousness of the role they can play in affecting their communities through computing and empower them to move beyond simply learning to code. Instead, we can ask them what they want to code and why they want to code it.[5]

By situating computing education in real-world contexts that matter to students, we can engage more people in computing, with all the benefits that affords the youth and to society. Though this may help to produce much-needed programmers, it will also produce computationally literate, problem-solving citizens.

Reducing the Barriers for Putting Computational Action into Practice

There are many challenges young learners face when trying to develop impactful computational solutions. Many of these can be attributed to the context of computing education itself—often taking place in traditional computing labs, which are far removed from students' everyday lives. However, with the growing proliferation of mobile and ubiquitous computing, there is the potential for rethinking and recontextualizing where and how students learn computing. Computing education can now be freed from the desk-bound screen and connected to students' lives and communities.

The ability to connect to the lives of students represents a fundamental shift in computing, opening up new avenues for young people to see their worlds as "possibility spaces," spaces in which they can ask questions and build solutions that address personally identified needs. However, in order to empower young people to build these solutions, we need to provide platforms and learning environments that reduce the barriers for them to quickly build and implement their designs. As one example, we developed App Inventor, a blocks-based programming language that allows learners to build fully functional mobile applications without the need to deal with complicated syntax.

Computational Action: A New Way of Framing Computing Education for Impact

The fundamental shift in the role computing can play in students' lives also requires us to critically reexamine the

This fundamental shift in the role computing can play in students' lives also requires us to critically reexamine the goals of CS education.

goals of CS education, particularly for K–12 students. The goal of computing education needs to move beyond computational thinking to a perspective of *computational action*. A computational action perspective on computing is founded on the idea that, while learning about computing, young people should have the opportunity to do computing in ways that have direct impact on their lives and their communities.

Through multiple design studies, workshops, and global mobile app development initiatives that used MIT App Inventor, we have developed two key dimensions for understanding and developing educational experiences that support students in engaging in computational action: computational identity and digital empowerment. *Computational identity* builds on prior research that showed the importance of young people's development of scientific identity for future STEM growth.[6] For us, computational identity is a person's recognition of themselves as capable of designing and implementing computational solutions to self-identified problems or opportunities. Further, the students should see themselves as part of a larger community of computational creators. *Digital empowerment* builds from the work of Freire[2], which situates empowerment as the ability to critically engage in issues of concern to them, and Thomas and Velthouse,[9] who see empowerment connecting to the concepts of meaningfulness, competence, self-determination, and impact. As such, digital empowerment involves instilling in young learners the belief they can put their computational identity into action in authentic and meaningful ways on issues that matter to them.

In order to develop computational action educational initiatives, we have developed a set of criteria that outline the critical elements required.

Supporting the formation of computational identity requires:

- Students must feel they are responsible for articulating and designing their solutions, rather than working toward predetermined "right" answers.
- Students need to feel their work is authentic to the practices and products of broader computing and engineering communities.

Supporting the formation of digital empowerment requires:

- A significant number of activities and development should be situated in contexts that are authentic and personally relevant.
- Students need to feel their work has the potential to make an impact in their own lives or their community.
- Students should feel they are capable of pursuing new computational opportunities as a result of their current work.

Computational Action in Action

We have seen firsthand the powerful effect a computational action approach can have to learning computer science. In the slums of Dharavi in Mumbai (one of the largest slums in Asia, and the iconic location of the film *Slumdog Millionaire*), a group of young women (8–16 years old) recognized women's safety was a critical problem in their community. Despite having no prior programming experience, they were driven by the feeling they could effect real change in the lives of those close to them. Through guidance from a local mentor, some online videos, and MIT's App Inventor, they were able to build the Women Fight Back app, which focuses on women's safety and has features like SMS alerts, location mapping, distress alarm, and emergency calls to contacts.[4] Inspired by their early success, these young women built several more apps, including one to coordinate water pickup from public water sources, and an educational app for girls who cannot go to school. These young learners' growth from no computing experience to a group that is continually working to improve their community through computing, shows the transformational potential computational action can have.

Building on the success of the Dharavi girls and other young learners like them, we have begun developing formal computing curricula that incorporate the computational action model. Recently, working with teachers at a large, extremely diverse, urban, U.S. high school, we created a 10-week computing curriculum with App Inventor. In this curriculum, students developed computing solutions to an issue that was personally relevant and meaningful to them and their community: raising awareness and cleaning up the local riverway. Exit interviews highlighted positive changes in the students' perceptions of their own computational identities and digital empowerment. From not believing themselves capable of building mobile apps at all, they realized they could not only build apps, but that their designs could have significant real-world impact. Many students also expressed excitement to build new apps in the future.

Facilitating this kind of learner-driven and action-focused computing education requires a reexamination of how we provide support for learners. It also poses new challenges for teachers. Students need scaffolding in the design process to help them understand how to decompose their apps into manageable and buildable parts. Importantly, teachers need to be comfortable in complex, real-world situations that do not have a predefined solution. While this should not require teachers to learn more about programming functionally, it will require them to be more flexible in how it is applied. It will require new strategies for helping students discover solutions on their own (rather than giving them the answer), and it will require new ways of assessing student work. Recognizing these pedagogical shifts means we must embrace new educational approaches as we test and refine our theories on computational action.

By focusing on computational action instead of computational thinking, we engage kids in meaningful projects rather than canned exercises.

Learners Recognize Opportunities to Apply Computing, then Design and Build Solutions

Having students drive their learning or problem-solving process is not a new idea in education. Problem-based learning (see for example Hmelo-Silver[3]) has been increasingly used in science and engineering education over the past two decades. However, putting the products students design into their communities has been a persistent challenge. Through the proliferation of mobile and ubiquitous computing, we are beginning to realize this potential.

By focusing on computational action instead of computational thinking, we engage kids in meaningful projects rather than canned exercises. Papert argued that in the process of developing personally meaningful projects, students would be able to forge ideas and would learn the necessary coding elements by addressing challenges as they naturally arise.[7] This is similar to how much programming and computational solution building works in the professional world. People from all occupations and avocations alike come up with "projects" they want to build for which computer programs are necessary. These people plan ahead and begin building, but inevitably, obstacles arise. These computer programmers, professionals and amateurs, computer scientists, engineers, scientists, and many others, find answers to those problems within the broader community of programmers (by asking colleagues directly or through sites such as StackOverflow). If this is the how computing happens in the real world, why is the educational system so often focused on students learning computing and computational problem solving in abstracted and inauthentic ways?

With rapid changes happening in both computing and computing education landscapes, we have an opportunity to reconsider how students learn computing. Young learners have the capacity to develop computational products that have authentic impact in their lives from the moment they begin to code. They simply need contexts that allow them to have such impact. Computational action starts to define what these contexts should look like. With more computing instructors coming online, we have a unique opportunity to work with them as they develop skills and practices necessary to engage in computational action with their students. We are excited about a world in which young learners see the world as full of opportunities for them to digitally create the future they (and we) want to inhabit.

References

1. Flegg, J., Mallet, D., and Lupton, M. Students' perceptions of the relevance of mathematics in engineering. *Intl. Journal of Mathematical Education in Science and Technology 43*, 6 (June 2012), 717–732.
2. Freire, P. *Pedagogy of the Oppressed* (20th anniversary ed.). Continuum, NY, 1993.
3. Hmelo-Silver, C.E. Problem-based learning: What and how do students learn? *Educational Psychology Review 16*, 3 (Mar. 2004), 235–266.
4. Joshi, S. Teenage girl coders from Mumbai slum are building apps to solve local problems. (Mar. 29, 2016); http://mashable.com/2016/03/29/mumbai-dharavi-girls-coding-apps/
5. Lee, C.H. and Soep, E. None but ourselves can free our minds: Critical computational literacy as a pedagogy of resistance. *Equity & Excellence in Education 49*, 4 (Apr. 2016), 480–492.
6. Maltese, A.V. and Tai, R.H. Eyeballs in the fridge: Sources of early interest in science. *International Journal of Science Education 32*, 5 (May 2010), 669–685.
7. Papert, S. An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning 1*, 1 (Jan. 1996), 95–123.
8. Pinkard, N. et al. Digital youth divas: Exploring narrative-driven curriculum to spark middle school girls' interest in computational activities. *Journal of the Learning Sciences 26*, 3 (Mar. 2017); doi.org/10.1080/10508406.2017.1307199
9. Thomas, K.W. and Velthouse, B.A. Cognitive elements of empowerment: An "interpretive" model of intrinsic task motivation. *Academy of Management Review 15*, 4 (Apr. 1990), 666–681.
10. Wing, J.M. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

Mike Tissenbaum (miketissenbaum@gmail.com) is Assistant Professor in the College of Education at the University of Illinois at Urbana-Champaign, IL, USA.

Josh Sheldon (jsheldon@mit.edu) is Associate Director, App Inventor, at MIT, Cambridge, MA, USA.

Hal Abelson (hal@mit.edu) is Class of 1922 Professor of Computer Science and Engineering in the Department of Electrical Engineering and Computer Science at MIT. Cambridge, MA, USA.