Received: 2 August 2020 | Accepted: 29 January 2021

DOI: 10.1111/bjet.13072

ORIGINAL MANUSCRIPT

British Journal of Educational Technology BERA

The case for alternative endpoints in computing education

Mike Tissenbaum[1] | **David Weintrop**[2] | **Nathan Holbert**[3] | **Tamara Clegg**[2]

[1]College of Education, University of Illinois Urbana-Champaign, Champaign, IL, USA

[2]Department of Teaching & Learning, Policy & Leadership, College of Education, College of Information Studies, University of Maryland, College Park, MD, USA

[3]Department of Mathematics, Science, and Technology, Teachers College, Columbia University, New York, NY, USA

Correspondence
Mike Tissenbaum, College of Education, University of Illinois Urbana-Champaign, Education Bldg, 1310 S 6th St, Champaign, IL 61820, USA.
Email: miketissenbaum@gmail.com

Funding information
NSF, Grant/Award Number: #1441184

Abstract

This paper argues for a re-examination of the nature and goals of broad computing education initiatives. Instead of starting with specific values or goals, we instead begin by considering various desired endpoints of computing instruction and then work backward to reason about what form learning activities might take and what are the underlying values and principles that support learners in reaching these endpoints. The result of this exercise is a push for re-thinking the form of contemporary computing education with an eye toward more diverse, equitable and meaningful endpoints. With a focus on the role that constructionist-focused pedagogies and designs can play in supporting these endpoints, we examine four distinct cases and the endpoints they support. This paper is not intended to encompass all the possible alternate endpoints for computer science education; rather, this work seeks to start a conversation around the nature of and need for alternate endpoints, as a means to re-evaluate the current tools and curricula to prepare learners for a future of active and empowered computing-literate citizens.

KEYWORDS

computing, constructionism, social inclusion, social justice

Practitioner notes

What is already known about this topic

- There is a growing call for computing education to be a core educational component.
- Computing education traditionally has a narrow goal of training people for programming jobs.
- Computing education fails to connect with students underrepresented in STEM.

What this paper adds

- An argument for why we need more and diverse endpoints to computing education.
- That many possible endpoints for computing education can be more inclusive, just and equitable than software engineering.
- Constructionism is a particularly useful paradigm for approaching and supporting alternate endpoints.

Implications for practice and policy

- Helps reframe the goals of computing education, to truly be "for all."
- Provides a set of cases for how this reframing can be achieved.
- Gives policy new lenses for understanding, evaluating and implementing computing education.

INTRODUCTION

In recent years, there has been a concerted effort to make computing and coding a core educational experience in countries throughout the world (eg, CSforAll). A variety of arguments have been given for these large-scale efforts including a desire to support young people in being able to "express themselves digitally" (BBC, 2019), seeking to empower youth to be able to impact their communities through programming (Bhattacharya, 2017) and wanting to provide learners with "the computational thinking skills they need to be creators in the digital economy" (Smith, 2016). Beyond simply learning to code, these goals of computing education—creative expression, social justice and economic opportunity—are frequently cited as primary reasons all youth should learn the powerful ideas of computing (Blikstein & Moghadam, 2018). In their review of motivations for bringing computing instruction into classrooms, Vogel and Colleagues (2017), introduced new goals for computer science instruction that included creating an informed citizenry and improving general technological literacy to the aforementioned objectives. The diversity of these goals highlights how computing has become a core part of society and point to a need for broadening what we consider to be desirable endpoints of computing education, to ensure they connect with all young people regardless of school, age, or interest.

Despite the growing chorus of voices pushing us to rethink the goals, values and priorities of contemporary computing education (Vakil, 2018), computing education has continued to center economic paradigms. The result is a landscape where the goal of most large-scale computing education initiatives is to funnel learners into a narrow pathway toward computer science degrees and jobs in the technology sector. This homogeneous model of computing education does a disservice to both the field of computing and to learners who miss out on opportunities for empowering, meaningful and lasting computational learning experiences.

In challenging this narrow view of computing education endpoints, we draw from the constructionist model set by Papert (1996), who wrote about reimagining mathematics learning and proposed Logo as an alternative form of mathematics education. Inspired by this work, we aim to identify just a few points in the expansive space of computing education that contrast in form and motivation from its current state. We start with the question "*what might people do with computing?*" and use this as a means for envisioning alternative computing endpoints that could and should exist in the landscape of computing education. In doing so, we argue that computing education should acknowledge and enact, the belief that computing has meaning and value in a host of potential contexts and daily experiences. Furthermore, opening up the space of "what counts" in computer science invites educators, policy makers, designers and learners to articulate their own vision of computing. The emergent nature of these new forms suggests a computing education model that can be more culturally responsive, critical and personally transformative.

Below, we outline why multiple endpoints for computing education are not only desirable, but also critical for supporting learners from all walks of life as they come to understand how computing can be a platform for reasoning about the world, for social change and for personal expression and empowerment. To ground this exploration of alternative computing education in authentic uses of computing, this paper lays out four distinct computer science endpoints outside of the conventional computer science pipeline. Accompanying each endpoint is an example of a learning experience to support learners along that pathway. In doing so, we show how the consideration of alternative endpoints and goals for computing education can reshape the tools used, the ways learners are supported in engaging with computer science ideas and a more equitable, inclusive version of computing education. Ultimately, reimagining the computing education landscape will productively change what it looks like for learners to authentically participate in meaningful computing and better prepare them to succeed in an increasingly computational world, for whatever definition of "success" they so choose.

MOTIVATIONS FOR CONSIDERING ALTERNATIVE COMPUTING ENDPOINTS

Visions for computer science education

Working with New York City school district stakeholders, Vogel et al. (2017) conducted a study seeking to catalog what they call "visions for computer science education." In total, they collected 161 arguments for computer science instruction and grouped them into seven categories: (a) economic and workforce development, (b) equity and social justice, (c) competencies and literacies, (d) citizenship and civic life, (e) scientific, technological and social innovation, (f) school improvement and reform and (g) fun, fulfilment and personal agency. This plurality of motivations is rarely reflected in the nature of the tools, activities and assessments used as part of computing education instruction. In fact, as learners' age, CS instruction becomes increasingly narrow with an explicit prioritization for college and career readiness. While this may be useful for some, it prepares learners for a narrow band of future careers and does little to address systemic issues of inequity in computing fields (Washington, 2020). The focus on coding for younger learners often fails to connect with their lived lives, visions of the jobs they want and the skills they may need to pursue their passions, creating a disconnect and withdrawal from computing education, particularly by those underrepresented in computing and STEM broadly (Valla & Williams, 2012). Importantly, a more expansive set of motivations appeals to a broader set of learners and can bring historically ignored voices, values and ideas into computing.

Alternative endpoints in pursuit of empowerment, civic engagement and equity

By focusing computing education on the iterative process of building functional and useful computational artifacts, learners have the opportunity to engage in critical reflection on what they are making and why and how it relates to them personally and society more broadly (Ratto & Boler, 2014). The consideration of the social, cultural and ethical dimensions of their constructions can serve as a pathway toward a deeper understanding of how computing shapes the world around them and their ability to create with it for their own goals, identity construction and expression (Lee & Soep, 2016; Vakil, 2018).

There is increasing recognition that the ability to read and write algorithms and computational systems is not just a skill for gaining employment, but is also crucially a civil rights issue (Noble, 2018; Vakil, 2018). Recent work has shown how machine learning and automated algorithmic approaches are deeply impacted by institutional racism (Nobel, 2018; Robinson et al., 2020), further articulating the need for a diversity of voices helping shape these digital innovations. In a society defined by computational systems, there is a need for all youth—particularly youth of color whose access to housing, education, food, safe recreational spaces, etc. has been systematically restricted in societies, such as the United States, founded on white supremacy—to understand how computing can be used as a means of personal empowerment and civic engagement (Benjamin, 2019). Computing education then must acknowledge not only the utility of code, but also invite learners to question who these computational systems serve (Soep & Lee, 2020). Likewise, learners should be invited to build systems that meet their needs and those of their community. Failing to support this connection between computing and the actual experiences of learners will result in computing education that marginalizes all but the dominant coding class (Margolis et al., 2008).

An alternate endpoints framing of computing education allows us to reimagine the purposes and forms of computing education toward addressing the structural inequalities and biases that exist within the field and have implications well beyond it. Reflecting on what computing education can be and framing it as a counter-narrative to what it currently is, provides the opportunity to directly address historical injustices and in doing so, draws on the knowledge, experience and perspectives of *all* learners. Legitimizing and valuing endpoints beyond conventional computer science careers can lead to more inclusive and welcoming forms of computing education, where creative, expressive and culturally valued instantiations of computing ideas are valued alongside the programming.

Culturally sustaining pedagogy

Considering alternative endpoints in computing education also serves as an opportunity to better align computing education efforts with culturally sustaining pedagogies (CSP) (Paris & Alim, 2014), which advocate the need to promote pluralistic engagement in learning. Specifically, CSP states that educators and learning environments should help learners not only see personal relevance in what they are learning, but they should also help them *maintain* and *engage* with their own cultural, linguistic and literate ways of being. This is particularly important for non-dominant youth who are often asked to put aside their own ways of being to perform in ways that resonate with dominant white American culture.

Culturally sustaining pedagogies emphasize three main critiques of asset-based approaches to learning that guide pluralistic approaches and experiences for learners, particularly those from non-dominant cultures/communities of color (Paris & Alim, 2014). First, educators and learning environments must help learners use their own (non-dominant) cultural, linguistic and literate practices, and also help them leverage diverse linguistic, cultural and literate norms, promoting their ability to flexibly navigate and engage within such plurality. Second, CSP educators and environments cannot assume one static culture, language, or literacy for any group of people, but instead must recognize that such practices and languages are always shifting and changing. Learning environments must adapt to and enable learners to incorporate their own dynamically evolving cultural, linguistic and literacy practices. Finally, CSP researchers point to the need for critical inward reflection on problematic aspects of youth and community cultures, as such practices are incorporated into learning environments (Ladson-Billings, 2014).

Applying CSP to the consideration of alternate endpoints for computing education is vital to ensure that computing is not only open and meaningful to learners with diverse experiences and perspectives, but also to build a future where computing can evolve and grow in response to shifting needs, trends, ideas and goals. It is important to acknowledge this shift extends beyond tools and curricula to include teachers as central to creating equitable, inclusive and effective computational learning experiences (Ryoo, 2019).

CONSTRUCTIONISM AS A FRAMEWORK FOR EXPANDING COMPUTING EDUCATION

We argue that the values and design practices of constructionism—that learning happens best when people are invited to collaboratively create artifacts that have personal and communal meaning (Holbert, Berlandet al., 2020; Papert & Harel, 1991)—should play a central role in reimagining what computing education that supports multiple endpoints might look like.

Throughout his writings on constructionism, Papert touted that children learn how to think critically through the process of solving problems that arise while programming computers (1980). Through the creative and investigative processes that are at the core of constructionism, learners begin to understand the multi-faceted ways that computing can be a central force for them to personally express themselves, construct their digital and personal identities, and empower them to be critically aware and empowered citizens. In placing the needs, goals and ideas of the learner at the center of the learning experience, constructionism can serve as a powerful lens for reimagining computing education both in terms of what types of activities learners engage in and the form and structure of the tools they use.

Constructionism's attention to learner's values and interests makes it well suited to support them in using computational power to explore a diverse range of experiences, practices, phenomena, etc. Whether supporting young people in constructing video games (Kafai, 1991), interactive art (Papert & Solomon, 1971), musical instruments (Gorson et al., 2017), e-fashion (Buechley & Eisenberg, 2008), or e-textiles (Fields et al., 2018), constructionists have cared deeply about supporting learners as they express their passions, explore their interests and work to design solutions to real-world problems. This foundational quality must be present in any effort to broaden participation in computing education. When learners are given the space to construct objects—either digital or physical—that have personal or communal meaning, they have the opportunity to represent these passions in inspectable artifacts that can be viewed, critiqued, extended, or repurposed by others. These opportunities arise in a myriad of ways including, but not limited to, maker faires, posting their own digital portfolios online and community sharing portals (such as the project gallery for sharing projects on the Scratch platform; or

GitHub, the massive open-source code-sharing website) that allow users to share, remix and reuse the work of others). This not only has important cognitive benefits—being able to externalize one's thinking into representational systems that can be debugged, modified, etc.—but also critical identity implications. Whether sharing digital work using online portfolios or projects galleries (such as on the Scratch platform or GitHub, the massive open-source code-sharing website), or physical work at maker faires or community makerspaces, the computer code and resulting artifact can serve as a representation of one's work and one's contributions to the broader computing community. From a constructionist perspective, computing is not just an economically viable way to make things, but a way of *doing* things, with others or for others, to change and impact the world.

ALTERNATIVE ENDPOINTS FOR COMPUTER SCIENCE

In this section, we introduce four distinct endpoints for computing education and present our work on creating pathways for learners to pursue them. The goal of this work is to argue for the importance of "computing for all," while also providing legitimate and authentic applications of computer science knowledge outside the existing pathways that prioritize careers in the technology sector as the end goal. This set of examples is by no means intended to be exhaustive, but instead is intended to demonstrate and highlight the diversity of endpoints and start a larger conversation around what computing education can be.

Endpoint: Impacting local communities and immediate needs

The first endpoint we consider is the development of novices' identities as empowered citizens capable of addressing real issues in their own lives, schools and communities with computing. While traditional computing education is often locked to desktop computers in computer labs or laptop carts in classrooms, the introduction of mobile technologies (smartphones in particular) has allowed computing education to move out of the classroom and into learners' everyday lives. This ability for the products that students create to be taken out of the computer lab and into the world has allowed students and educators to move beyond simply writing code and instead critically asking *why* and *who* they are building it for, and to what end (Holbert, 2016; Lee & Soep, 2016). By situating computing education directly in students' lives, we open up computing education as a possibility space for impact and empowerment. This is critically important, as a long line of research has shown that the failure to meaningfully connect computing to the personal lives of students contributes to learners feeling computing is not useful or relevant to them (Couragion Corporation, 2018; Margolis & Fisher, 2003). This is particularly true for students underrepresented in computing and engineering careers (Pinkard et al., 2017).

In response, we conducted a study based on the ideas of *computational action* (Tissenbaum et al., 2019), which focuses on three key factors: (a) *computational identity*, which is a person's recognition of themselves as capable of identifying and creatively implementing computational solutions to issues in their lives, schools and communities; (b) *digital empowerment*, which focuses on people's ability to put their computational identity into action in authentic and personally meaningful ways; and (c) *computational design thinking*, in which learners' can successfully articulate the processes by which they will design and develop their solutions.

In order to support students' engagement in computational action, we need tools that reduce the barriers for them to quickly build, implement and refine their designs. One example of a platform particularly well-suited for this is MIT's App Inventor, a block-based programming language that enables users to build fully functioning, native Android mobile applications. However, it is not enough to provide novice learners with a coding platform and simply let them loose. Supporting computational action also requires the development of scaffolds in the form of support materials (such as design documents) and scripted activities that lead students through the design process. Developing these additional supports is a key to ensuring that students progress from ideation to implementation.

To explore how a computational action-focused curriculum can support students in developing meaningful solutions to personally relevant issues, Tissenbaum et al. (2019) implemented a computational action curriculum in an ethnically diverse urban high school in the United States. Together with the teacher, the project team identified an issue that was of interest to students at the school and the broader local community: the pollution of the local river (a major feature that runs through the city). Working in collaborative teams, students developed their own solutions to increase community awareness and strategies for cleaning up the river. To provide students with an opportunity to present their work in authentic contexts (ie, to support their *digital empowerment*), students were invited to present their final projects at the school-wide job fair, which included visits from local city council members and the mayor.

At the end of the curriculum, many of the students expressed that at the outset they never thought they would be able to build an app themselves, let alone build one that they felt had the potential to make real change. Many expressed excitement toward developing solutions to new problems using the computational tools and knowledge developed during this project.

As this example shows, a computational action approach to computing education has the potential to support students to become, not only programmers but computationally literate, empowered problem-solving citizens.

Endpoint: Data literate athletes and healthy citizens

The next endpoint we consider is CS for data literacy. While data literacy is often included as part of K-12 computer science standards and curricula (eg, K12cs.org, 2016), the focus on data literacy in CS often emphasizes programming aspects related to data—eg, data structures, simulations/models and data sources. Definitions of data literacy could be broadened to consider aspects from the fields of Information and Data Science (Clegg et al., 2020). From Information Science, data literacy involves knowledge about how to access, interpret, critically assess, manage, handle, communicate, preserve and ethically use data to gain a new understanding of the world (Maybee & Zilinski, 2015; Prado & Marzal, 2013). In Data Science, data literacy also includes practically applying insights from data to real-world decisions (eg, Provost & Fawcett, 2013). Taking this broadened definition of data literacy as an endpoint to CS provides extensive, felicitous personal connections to health, sports and athletics—life-relevant domains for many learners who are deeply engaged in sports and health contexts (Clegg et al., 2017).

We have applied this endpoint in two contexts. First, Clegg et al., (2020) are conducting a study with Division 1 collegiate athletes and athletics staff to understand the data literacy practices athletes (as well as coaches and other athletics staff) are already engaging in. Mounds of data are collected about the performance and training of athletes. Athletes, therefore, have to learn how to narrow their focus to the most actionable data, often requiring the assistance of coaches and staff to determine the most relevant stats and analyses to consider at any given time (Clegg et al., 2020). They must then evaluate their data (and their opponents') to adjust their training and regimen for peak performance. This activity of organizing, analyzing, visualizing and interpreting data relies on computational knowledge, practices and tools. These findings suggest the potential for leveraging sports contexts to support data literacy learning experiences as an alternative endpoint for CS.

A computing endpoint focused on data literacy can also extend beyond sports-specific experiences and focus on health more generally, particularly emphasizing quantified-self experiences for learners (Lee et al., 2016). We have specifically explored such health experiences through the development of *live-physiological sensing and visualization (LPSV)* tools for elementary-aged learners in the *BodyVis* project (Clegg et al., 2017). Children wear sensors that capture their heart-rate and breathing rate in real-time and draw evidence-based conclusions about their body through analysis of visualizations of their live body data as displayed on an e-textile shirt (Norooz et al., 2015, Figure 1a,c) and a large-screen display (Kang et al., 2016, Figure 1b,d) (Clegg et al., 2017). These tools provide opportunities for learners to analyze and reason with data (eg, determining how everyday activities like eating, dancing, or doing homework affect their heart rate). More recent work has extended this approach to enable children to design, program and build their own wearable models of their circulatory systems using programmable sensors and e-textiles (Figure 1a,c).

Similar to other endpoints, this approach requires scaffolding in the form of curricula, tools and facilitation. For example, we have found that early elementary-age learners need specific constraints for designing investigations with LPSV tools (eg, experiments must be done within one minute, inside the classroom and with BodyVis or SharedPhys) as well as vocabulary words to help them understand relevant terms (eg, increase, decrease) (Byrne et al., 2018). With support in place, elementary-aged learners have been able to design their own personally relevant investigations with data, draw meaningful conclusions, as well as design, program and build their own simulations based on what they learn with sensor-based tools. Taken together, these health and sports studies show the significant potential for leveraging data literacy as an alternate CS endpoint that offers rich epistemological and personal connections for learners.

Endpoint: Means of personal and social creative expression

Many computing initiatives and tools have had as their goal supporting young people to express themselves digitally. Logo, the first programming language for children, was often used to create "computer graphics" similar to those seen at the arcade and on the video game systems of that time (Harel & Papert, 1990). With Scratch, children can create interactive stories, games and animations (Resnick et al., 2009). Similarly, the so-called maker movement invites people of all ages to build personally interesting tangible artifacts that combine computing with fabrication and craft work (Halverson & Sheridan, 2014). All of these efforts see the computer and code as a digital canvas, a medium that enables a host of alternative forms of creative expression.

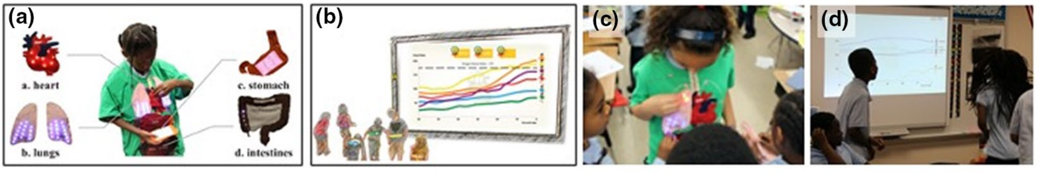


FIGURE 1 BodyVis (a) shows learners' live heart and breathing rates on an e-textile shirt and SharedPhys (b) visualizes live heart rate data of up to 6 wearers on a moving line graph shown on a large display. Learners engage in collaborative classroom experiences with BodyVis (c) and SharedPhys(d). (Clegg et al., 2017) [Colour figure can be viewed at wileyonlinelibrary.com]

Though many computing education efforts—especially those for younger learners—invite learners to make aesthetically interesting artifacts, these activities are often used as a means to the end of learning specific computing content knowledge or practices. However, creative construction offers more than just a compelling way to encounter the practices of the software engineer (Peppler, 2010). Though the design and creation of compelling artifacts that speak to the experiences, values, or perspectives of society have traditionally been considered the domain of the artist, here we examine how creative expression itself can be a powerful and worthwhile endpoint of computing education.

In the Remixing Wakanda project, code and fabrication became the materials with which to imagine and create futuristic artifacts and societies that challenge and critique the current state of the world (Holbert, Dandoet al., 2020). Rather than design artifacts or technologies that exist within the space of the probable, in the Remixing Wakanda project, five Black teenage girls working with professional comic book artists, learning scientists, designers and local activists, used the design genre of Afrofuturism (Anderson & Jones, 2015) to bring into being futuristic technologies that represented their values and perspectives.

In these constructions, participants used computational tools, sensors and circuitry to creatively merge aesthetic considerations with functionality to highlight humanity's problematic relationship with the environment and to acknowledge and respond to their experiences with racism and inequality (Figure 2). One participant used illustration software and a laser cutter to build a massive city where different communities and societies live together in harmony with nature. Another, reflecting on her personal frustration about litter and trash in the city, designed an aesthetically appealing trash receptacle that directly converts trash to the energy that can be used to power street lights (a safety concern for those that work late).

While some of these artifacts were hopeful visions of the future—they also reflected the learners' pasts and critiqued the inequities of the present (Holbert, Dando et al., 2020). For example, one participant designed a fashionable cloak that hid the wearer from prying eyes—eyes that she said, "make you feel like you're alien [...] like [you're] some art exhibit or something." Using textiles and patterns from her native Senegal, the cloak elevated a distinctly African aesthetic. It also included a battery of sensors that monitored the health and wellbeing of the wearer. Together the cloak represented not just the history and experience of this young woman, but also her need to protect herself from the racist gaze she experienced as a young Black woman in a large American city, as well as the physical harm

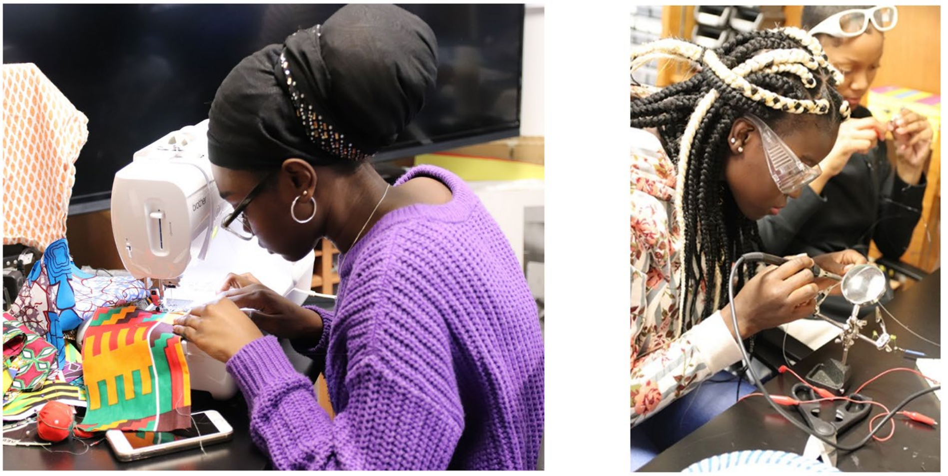


FIGURE 2 Learners constructing artifacts as part of the Remixing Wakanda project [Colour figure can be viewed at wileyonlinelibrary.com]

perpetuated through actual physical violence, as well as systemic inequality against those that look like her.

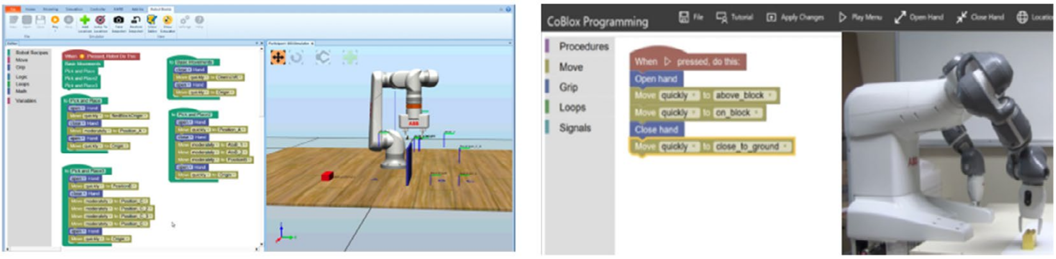
In each case of the Remixing Wakanda project, computing is a tool for critically reflecting on the current state of the world and for creating representations of a possible future that might initiate change today. While participants did encounter coding and potentially came away with new knowledge about computing concepts or practices, these experiences are themselves a means of creative expression, of creating artifacts and representations that center their anxieties and fears as well as their hopes and dreams. In the Remixing Wakanda project, computing is a tool for engaging in critical reflection on inequitable societies, unsustainable energy practices and systems of oppression. Here, computing is a means of agitating for change—itself a powerful and important endpoint.

Endpoint: Blue collar computing

Our final example of an alternative endpoint for computing education is motivated by economic concerns like many efforts to broaden CS education, but is different in the type of jobs for which it is seeking to prepare learners. Whereas current computing education is focused on preparing learners to meet the growing demand in the technology sector (eg, software development), there is an opportunity and need for computing education to prepare learners for positions of all kinds that depend on expressing ideas and instructions in computationally meaningful ways (Thompson, 2017). Considering such endpoints in our thinking about the purpose and goals of computing education reflects a number of important realities. First, not all learners will become software engineers so we should not treat software development as the only profession where such skills are useful. Second, the economy needs a broad range of workers, thus computing education should reflect the plurality of professional needs. Finally, the nature of work is changing, technology is creating new types of jobs and introducing new demands on workers (Wilson et al., 2017). Having a foundational understanding of computing can help prepare learners to work with and alongside computing in newly created or reimagined sectors.

For example, the nature of so-called "blue collar work" or manual labor jobs is shifting dramatically in response to increasing automation and robotics in manufacturing (Autor, 2015). For example, when robots are introduced into manufacturing, they usually only perform a subset of the required actions, resulting in robots and humans working side-by-side complementarily (Colgate et al., 1996). While robots excel at repetitive tasks that require a high degree of precision, humans are better at tasks that involve decision-making, adaptation and creativity (Blank et al., 2006). As such, humans working alongside such robots may be tasked with modifying existing robot routines or defining new sets of instructions to adjust to new requirements or constraints. Such a position requires the same set of foundational computing knowledge and skills as other high-tech jobs, but as the use of these skills is different, the tools that would accompany such an endpoint look different. Toward this end, we conducted research looking to redesign industrial programming interfaces so as to make them more accessible and intuitive for those new to computing.

CoBloX (Weintrop et al., 2018) is a block-based programming environment designed to allow users with little or no prior programming experience to control a one-armed industrial robot (Figure 3). Beyond the basic block-based interface, CoBloX also includes additional features to support programming industrial robots, including domain-specific programming commands, a virtual execution environment and predefined templates for common robot routines (Weintrop et al., 2017). In a comparison study of CoBloX alongside industry-standard programming interfaces, CoBloX was found to be easier to use and easier to learn, while helping adult novices complete robotics programming tasks faster without any loss in accuracy (Weintrop et al., 2018).



(a) (b)

FIGURE 3 Virtual (a) and physical (b) implementations of the CoBloX programming Environment [Colour figure can be viewed at wileyonlinelibrary.com]

The goal of presenting CoBloX as part of the case for alternative endpoints in computing education is to showcase what it looks like for computer science knowledge to be used in a professional setting historically not considered within the purview of computer science. In documenting the ways that computer science knowledge can be professionally employed beyond the technology industry, we show the importance of a broader vision of what computer science is and who it is for. With CoBloX, we see an authentic and legitimate professional endpoint in which computer science knowledge is valued but not typically included in the narrative around why computer science is important and one that may be important for learners who do not wish to pursue a career as a professional software developer.

DISCUSSION

The increasingly digital nature of our world requires that all learners understand the foundational ideas of computing and feel empowered to meaningfully participate in computational spaces. The last decade has seen the computer science community lead the designing of tools, creating curricula and crafting policy that shape this instruction. While it is important for computer scientists to have a seat at the table, it is equally important that the ideas, values and goals of those beyond the field are also present to reflect the growing role of computing in the world and not perpetuate existing disparities and injustices in the field. This is particularly true for women and students of color, who have been marginalized in computing education and have had to learn computing along racialized, gendered and class-influenced learning pathways (Margolis et al., 2008).

By envisioning and valuing alternative, yet equally valid and important, endpoints, this work seeks to start a conversation around the nature of and need for alternate forms of computing education. We see the ideas and principles of constructionism as having much to contribute toward realizing new forms of computing education that more fully reflect the plurality and diversity of possible computing endpoints.

Prioritizing economic outcomes in the motivation for computing instruction does not accurately reflect the ubiquitous role that computing plays in society. It ignores equally valid motivations for computing grounded in issues of equity, empowerment, expression and justice. By overly focusing on existing code-centric endpoints of computing education, we risk cementing current computer science education as a "sealed truth" rather than a space that needs to evolve and be contested as a culturally mediated set of knowledge and practices that are deeply enmeshed with the human experience (Paris & Alim, 2014). In order for CS education to truly be *for all* in ways that are representative, equitable and just, we need to embrace the broadest definition of computing endpoints as possible.

In order to achieve this vision, we also need to reconsider the environments in which learners construct both their understandings about computing and the products of their efforts. To this end, a constructionist design paradigm can also inform how to think about the structure and form of computational tools and environments that can support learners as they express themselves, explore computing in personally relevant ways and use computing to make meaningful change in the world.

The examples given in this paper are not meant to be exhaustive in terms of the range of new possible endpoints, rather they offer a vision of how we can reconceptualize both the goals for computing and environments that can support learners in realizing them. Other constructionist approaches could be similarly seen to advocate for alternate endpoints, such as work on physical computing as a pathway toward computational expression (Martin et al., 2000), the blending of traditional crafting practices with computing (Buechley & Eisenberg, 2008) and using computing and programming as a way to hold a critical lens up to inequities, power and injustice in society (Soep & Lee, 2020). Constructionist designs have also helped bring multimedia constructions, animation and story-telling into computing education spaces (Resnick et al., 2009). Collectively, the constructionist design approach has played a central role in reimagining the tools that learners use to explore the powerful ideas of computing, imagine new futures and define what computing and computing education mean for them. By embracing a vision of broader computing endpoints, we can excite learners their potential to be active and empowered designers and creators of our digitally mediated futures.

ACKNOWLEDGMENTS

This work benefitted from several funding sources, including ABB Robotics. This paper expands on ideas previously published in Weintrop, D., Holbert, N. & Tissenbaum, M. (2020). Considering alternative endpoints: An exploration in the space of computing educations. In *Proceedings of Constructionism 2020 Conference* (pp. 213–223), Dublin, Ireland.

CONFLICTS OF INTEREST

There are no conflicts of interest with any of the work presented by any of the authors.

DATA AVAILABILITY STATEMENT

As this paper is largely theoretical, there is not primary data to link to. Readers who are interested in the data of the described studies can find it in the papers references herein or contacting the authors directly.

ETHICS

All of the work described in this paper was done under the approval of each institutions' respective Institutional Review Board (IRB).

ORCID

Mike Tissenbaum https://orcid.org/0000-0003-0356-5448

REFERENCES

Anderson, R., & Jones, C. E. (2015). *Afrofuturism 2.0: The rise of astro-blackness*. Lexington Books.

Autor, D. H. (2015). Why are there still so many jobs? The history and future of workplace automation. *The Journal of Economic Perspectives*, *29*(3), 3–30. https://doi.org/10.1257/jep.29.3.3.

BBC. (2019). *Make it digital—The BBC micro:bit*. BBC. https://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit

Benjamin, R. (2019). *Race after technology: Abolitionist tools for the new Jim code*, (1st ed.). Polity.

Bhattacharya, A. (2017). *What happens when girls in one of the world's largest slums start coding and building apps*. Quartz India. https://qz.com/india/1032018/dharavi-diary-what-happens-when-girls-in-one-of-the-worlds-largest-slums-start-coding-and-building-apps/

Blank, D., Kumar, D., Meeden, L., & Yanco, H. (2006). The Pyro toolkit for AI and robotics. *AI Magazine*, *27*(1), 39.

Blikstein, P., & Moghadam, S. H. (2018). *Pre-college computer science education: A survey of the field*. Google LLC. https://goo.gl/gmS1Vm

Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: Toward wearable engineering for everyone. *Pervasive Computing, IEEE*, *7*(2), 12–15. https://doi.org/10.1109/MPRV.2008.38.

Byrne, V., Kang, S., Norooz, L., Velez, R., Katzen, M., Ade, A., Froehlich, J., & Clegg, T. (2018). Scaffolding authentic scientific inquiry experiences for early elementary learners using wearable technology. In J. Kay & R. Luckin (Eds.), *Rethinking learning in the digital age: Making the learning sciences count: Proceedings of the 17th International Conference of the Learning Sciences (ICLS'18)*. International Society of the Learning Sciences.

Clegg, T., Greene, D., Beard, N., & Brunson, J., (2020). Data everyday: Data literacy practices in a division I sports context. In *Proceedings of SIGCHI Human Factors in Computing Systems (CHI, 2020)*. ACM.

Clegg, T., Norooz, L., Kang, S., Byrne, V., Katzen, M., Valez, R., Plane, A., Oguamanam, V., Outing, T., Yip, J., Bonsignore, E., & Froehlich, J. (2017). (2017) Live physiological sensing & visualization ecosystems: An activity theory analysis. In *Proceedings of the 2017 SIGCHI Conference on Human Factors in Computing Systems (CHI'17)* (pp. 2029–2041). ACM.

Colgate, J. E., Edward, J., Peshkin, M. A., & Wannasuphoprasit, W. (1996). Cobots: Robots for collaboration with human operators. In *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, Atlanta, GA.

Couragion Corporation. (2018). *Altering the vision of who can succeed in computing*. Oracle Academy.

Fields, D. A., Kafai, Y., Nakajima, T., Goode, J., & Margolis, J. (2018). Putting making into high school computer science classrooms: Promoting equity in teaching and learning with electronic textiles in exploring computer science. *Equity & Excellence in Education*, *51*(1), 21–35.

Gorson, J., Patel, N., Beheshti, E., Magerko, B., & Horn, M. (2017). TunePad: Computational thinking through sound composition. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 484–489).

Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495–504. https://doi.org/10.17763/haer.84.4.34j1g68140382063.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments*, *1*(1), 1–32. https://doi.org/10.1080/1049482900010102.

Holbert, N. (2016). Leveraging cultural values and "ways of knowing" to increase diversity in maker activities. *International Journal of Child-Computer Interaction*, *9–10*, 33–39. https://doi.org/10.1016/j.ijcci.2016.10.002

Holbert, N., Berland, M., & Kafai, Y. (2020). 50 Years of Constructionism. In N. Holbert, M. Berland, Y. Kafai (Eds,) *Designing constructionist futures: The art, theory, and practice of learning designs*. MIT Press.

Holbert, N., Dando, M. B., & Correa, I. (2020). Afrofuturism as critical constructionist design: Building futures from the past and present. *Journal of Learning, Media, and Technology.*, https://doi.org/10.1080/17439884.2020.1754237

K–12 Computer Science Framework. (2016). http://www.k12cs.org.

Kafai, Y. B. (1991). Learning design by making games. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 71–96). Ablex Publishing Corporation.

Kang, S., Norooz, L., Oguamanam, V., Plane, A., Green, A., Clegg,T., Froehlich, J. (2016). SharedPhys: Live physiological sensing, whole-body interaction, and large-screen visualizations to support shared inquiry experiences. In *Proceedings of the 15th International Conference on Interaction, Design, and Children (IDC'16)* (pp. 275–287). : ACM.

Ladson-Billings, G. (2014). Culturally relevant pedagogy 2.0: A.k.a. the remix. *Harvard Educational Review*, *84*(1), 74–84. https://doi.org/10.17763/haer.84.1.p2rj131485484751

Lee, C. H., & Soep, E. (2016). None but ourselves can free our minds: critical computational literacy as a pedagogy of resistance. *Equity & Excellence in Education*, *49*(4), 480–492. https://doi.org/10.1080/10665684.2016.1227157

Lee, V. R., Drake, J. R., & Thayne, J. L. (2016). Appropriating quantified self technologies to support elementary statistical teaching and learning. *IEEE Transactions on Learning Technologies*, *9*(4), 354–365. https://doi.org/10.1109/TLT.2016.2597142.

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. The MIT Press.

Margolis, M., Estrella, R., Goode, J., Jellison Holme, J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. MIT Press.

Martin, F., Mikhak, B., Resnick, M., Silverman, B., & Berg, R. (2000). To mindstorms and beyond. In *Robots for kids: Exploring new technologies for learning* (pp. 9–36). Morgan Kaufmann.

Maybee, C., & Zilinski, L. (2015). Data informed learning: A next phase data literacy framework for higher education. *Proceedings of the Association for Information Science and Technology 52*(1):1–4. https://doi.org/10.1002/pra2.2015.1450520100108.

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

Norooz, L., Mauriello, M. L., Jorgensen, A., McNally, B., & Froehlich, J. E. (2015, April). BodyVis: A new approach to body learning through wearable sensing and visualization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp 1025–1034).
Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.
Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, *1*(1), https://doi.org/10.1007/BF00191473
Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism 36*(2):1–11.
Papert, S., & Solomon, C. (1971). *Twenty things to do with a computer*. http://18.7.29.232/handle/1721.1/5836
Paris, D., & Alim, H. S. (2014). What are we seeking to sustain through culturally sustaining pedagogy? A loving critique forward. *Harvard Educational Review*, *84*(1), 85–100. https://doi.org/10.17763/haer.84.1.982l873k2ht16m77.
Peppler, K. A. (2010). Media arts: Arts education for a digital age. *Teachers College Record*, *112*(8), 2118–2153.
Pinkard, N., Erete, S., Martin, C. K., & de Royston, M. M. (2017). Digital youth divas: Exploring narrative-driven curriculum to spark middle school girls' interest in computational activities. *Journal of the Learning Sciences*, *26*(3), 477–516. https://doi.org/10.1080/10508406.2017.1307199
Prado, J. C., & Marzal, M. Á. (2013). Incorporating data literacy into information literacy programs: Core competencies and contents. *Libri*, *63*(2), 123–134. https://doi.org/10.1515/libri-2013-0010
Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big Data*, *1*(1), 51–59. https://doi.org/10.1089/big.2013.1508.
Ratto, M., & Boler, M. (2014). *DIY citizenship: Critical making and social media*. MIT Press.
Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779.
Robinson, W. R., Renson, A., & Naimi, A. I. (2020). Teaching yourself about structural racism will improve your machine learning. *Biostatistics*, *21*(2), 339–344.
Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education*, *19*(4), 1–23. https://doi.org/10.1145/3322210
Smith, M. (2016). *Computer science for all*. Whitehouse.Gov. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all
Soep, L., & Lee, C. (2020). Code for what? In H. Jenkins, G. Peters-Lazaro, & S. Shresthova (Eds.), *Popular culture and the civic imagination: Case studies of creative social change*. NYU Press.
Thompson, C. (2017). The next big blue-collar job is coding. *Wired Magazine*, *24*(12).
Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, *62*(3), 34–36. https://doi.org/10.1145/3265747.
Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, *88*(1), 26–52. https://doi.org/10.17763/1943-5045-88.1.26.
Valla, J. M., & Williams, W. (2012). Increasing achievement and higher-education representation of under-represented groups in science, technology, engineering, and mathematics fields: A review of current K-12 intervention programs. *Journal of Women and Minorities in Science and Engineering*, *18*(1), 21–53. https://doi.org/10.1615/JWomenMinorScienEng.2012002908.
Vogel, S., Santo, R., & Ching, D. (2017). Visions of computer science education: unpacking arguments for and projected impacts of CS4All initiatives. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE'17* (pp. 609–614).
Washington, A. N. (2020). When twice as good isn't enough: The case for cultural competence in computing. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 213–219). https://doi.org/10.1145/3328778.3366792
Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D. C., & Franklin, D. (2018). Evaluating CoBlox: A comparative study of robotics programming environments for adult novices. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, *366*, 1-12. https://doi.org/10.1145/3173574.3173940
Weintrop, D., Shepherd, D. C., Francis, P., & Franklin, D. (2017). Blockly goes to work: Block-based programming for industrial robots. *IEEE Blocks and beyond Workshop*, *2017*, 29–36. https://doi.org/10.1109/BLOCKS.2017.8120406
Wilson, H. J., Daugherty, P. R., & Morini-Bianzino, N. (2017). The jobs that artificial intelligence will create. *MIT Sloan Management Review*, *58*(4), 14–16.

How to cite this article: Tissenbaum, M., Weintrop, D., Holbert, N., & Clegg, T. (2021). The case for alternative endpoints in computing education. *British Journal of Educational Technology*, 52, 1164–1177. https://doi.org/10.1111/bjet.13072