A New Age in Tangible Computational Interfaces for Learning

Paulo Blikstein, Stanford University, 520 Galvez Mall, Stanford, CA, 94305, paulob@stanford.edu (Organizer and Chair)
Leah Buechley, MIT Media Lab, 77 Massachusetts Ave., Cambridge, MA 02139, leah@media.mit.edu
Michael Horn, Northwestern University, 2120 Campus Dr., Evanston, IL, 60202, michael.horn@northwestern.edu
Hayes Raffle, Nokia Research Center, 955 Page Mill Road, Palo Alto, CA 94304, hayes.raffle@nokia.com

Discussant: Edith Ackermann, MIT School of Architecture, 77 Massachusetts Ave., Cambridge, MA 02139, edith@media.mit.edu

Abstract: Tangible computational technologies for education have evolved from research prototypes to the mainstream, led by the wide adoption of commercial kits such as the Lego Mindstorms platform. Despite their success, researchers have pointed out four main issues that have limited their adoption. First, most toolkits were marketed for a particular gender and age group. Second, they had standardized parts and proprietary connectors, which limited the types of projects that users could build. Third, they had programming environments with limited features. Lastly, they were difficulty to connect with school curricula in science and mathematics, which segregated robotics to after-school activities. In this symposium, we will: (a) investigate these limitations in light of current research in the field; (b) showcase several new designs for tangible computational artifacts for education which try to overcome these limitations; (c) discuss possible implications of the widespread use of these new computational artifacts.

Overview of Symposium Panel and Demonstrations

Over the past 15 years, tangible computational technologies for education, inspired by the constructionist tradition (Papert, 1980), have evolved from prototypes in research laboratories (for example, Martin, 1993; Resnick, et al., 1998; Resnick, Ocko, and Papert, 1991) to the mainstream, led by the wide adoption of commercial kits such as the Lego Mindstorms platform, the VEX Robotics kits, among several others. Despite the success of this first generation of digital manipulatives, particular design decisions had to be made to make them viable. First, most of these toolkits were marketed for middle-school boys (mostly robotics kits), which limited their use by young women. Second, their standardized parts and proprietary connectors made it easier for users to get started and backgrounded much of the complexity in building mechanical structures, but limited the types of projects that more advanced users could build, and made the integration with traditional materials difficult (textiles, paper, crafts, cheap electronics, scrap materials). Third, the limitations of extant programming environments and the intrinsic difficulties of mimicking complex physical behaviors with computer code also narrowed what students could accomplish, and made the activities especially hard for younger learners. Lastly, the difficulty in connecting the activities with school curricula in science and mathematics segregated the use of computational manipulatives to after-school activities.

The authors in this symposium are representative of a new generation of designers of computational tangibles for children. Each of the papers will discuss one novel emergent design framework which addresses many of current design issues in existing digital manipulatives. Michael Horn will discuss limitations of on-screen-only programming interfaces for children and show new designs in tangible programming artifacts, with which children can program using physical blocks. Paulo Blikstein will discuss the difficulties in connecting computational artifacts with scientific topics in physics and chemistry, and present the *bifocal modeling* platform, which enables learners to build their own scientific lab, collect empirical data, and match them to data from their own computer models. Leah Buechley will describe her work creating computational platforms out of paper and textiles, and how this has blurred the boundaries between traditional and computational media, and made tangibles less directed to just one gender. Finally, Hayes Raffle will show how his work with programmable-by-example computational tangibles ("Topobo") made them more approachable to younger audiences, and enabled children to program complex physical behaviors by constructing and moving robotic creatures.

These four emergent design frameworks are pointing to new directions in the use of computational tangibles in education:

1) **A wider palette of materials:** more materials and building techniques are being made available to children. In particular, low-tech materials with which students are already familiar can now be platforms for computation, as well as a wider selection of sensors, probes, and actuators.

2) **More diverse projects:** breaking away from the tradition of the gender-biased uses of robotic technologies to make robots and cars, these platforms allow children an entirely new array of expressive

possibilities, since the toolkits are composed of much more flexible and customizable parts. Projects such as interactive art, "animals" with realistic motion, sensor-enabled prototypes, scientific inquiry apparatus, and electronics sketchbooks are made possible and technically more approachable.

3) **More flexible programming modes:** traditional programming for computational tangibles has been based on text or block-based coding. The new platforms presented in the symposium enable for much more diverse modes of programming. Students can use "smart parts" that can remember motion (thus children can program a creature by example), physical smart blocks which can be combined together to create a program, or real-world sensor data for code optimization.

These three novel directions, as the individual papers will discuss, could point to a new age in the use of computational manipulatives for learning. Some of these technologies have already been tested in schools or after-school environments, but we believe that their collective presence in a symposium will enable researchers to have a more comprehensive view of where the field is going, and allow for rich discussions within the research community. In the symposium, authors will talk about their latest designs and research findings, and also do demonstrations of the actual devices and technologies.

Abstracts of Panel Participants

Topobo: programming by example to create complex behaviors

Hayes Raffle

Topobo is a 3D constructive assembly system embedded with kinetic memory—the ability to record and playback physical motion. Unique among modeling systems is Topobo's coincident physical input and output behaviors. By snapping together a combination of passive (static) and active (motorized) components, users can quickly assemble dynamic biomorphic forms like animals and skeletons with Topobo, animate those forms by pushing, pulling, and twisting them, and observe the system repeatedly play back those motions. For example, a dog can be constructed and then taught to gesture and walk by twisting its body and legs. The dog will then repeat those movements and walk repeatedly.

Figure 1. A Topobo Moose (left): to program motions, you just manipulate the toy; and Topobo pieces (right).

Topobo is a class of tools that helps people transition from simple-but-intuitive exploration to abstract-and-flexible exploration. The system is designed to facilitate cognitive transitions between different representations of ideas, and between different tools. A modular design approach, as well as an inherent grammar, helps people make such transitions. With Topobo, children use enactive knowledge, e.g. knowing how to walk, as the intellectual basis to understand a scientific domain, e.g. engineering and robot locomotion. Queens, Backpacks, Remix and Robo add various abstractions to the system, and extend the tangible interface. Children use Topobo to transition from hands-on knowledge to theories that can be tested and reformulated, employing a combination of enactive, iconic and symbolic representations of ideas.

In the past, systems for children to model behavior have been either intuitive-but-simple, (e.g. curlybot, Frei, 2000) or complex-but-abstract (e.g. LEGO Mindstorms). In order to develop a system that supports a user's transition from intuitive-but-simple constructions to constructions that are complex-but-abstract, I draw upon constructivist educational theories, particularly Bruner's theories of how learning progresses through enactive then iconic and then symbolic representations. Bruner (2004), after Piaget (1976, Cole & Cole, 2001), described a sequence of stages all people seem to progress through as they represent and acquire knowledge, moving from enactive, to iconic to symbolic representations of knowledge (Figure 2). Bruner's framework suggests that certain ideas can be made even more accessible, and at a younger age, if they can be grasped and manipulated physically. In this work, I show how tangible programming and interaction can provide an enactive mode of interacting with computers, where tangibles provide a bridge from computers' iconic and symbolic representations to enactive ones, and allow for more intuitive expression and access to certain ideas. My hypothesis is that physical, and especially spatial or 3D problems, are best approached first in the tangible domain, where simple behaviors can be prototyped and manipulated tangibly.

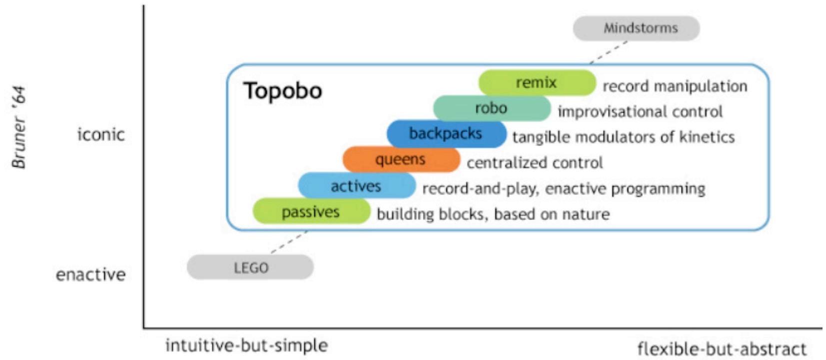

Figure 2. Sequence of stages of representation and acquisition of knowledge

In presenting the design and development of Topobo, I will talk about how more than 100,000 children have used the system through workshops and outreach, and how commercialization of the technology is putting tangibles for learning into young children's hands worldwide, helping them to climb a mountain of ideas about technology, robotics and the natural world.

LilyPad Arduino: rethinking the materials and cultures of educational technology

Leah Buechley

The LilyPad Arduino is a construction kit that enables students to construct and program tangible interactive devices (Buechley, 2008). Similar to Lego Mindstorms, it consists of a set of controllable input and output pieces like temperature sensors, light sensors, motors, and LEDs, but users of the LilyPad build interactive textiles instead of robots. Soft, wearable artifacts are made by stitching sewable components together with silver-plated, electrically conductive thread. Figure 3 shows a picture of the kit and a sample design, a jacket with turn signals on its back that was designed for cycling.

Figure 3. Components of the LilyPad Arduino (left) kit and a sample construction (right).

In an ongoing series of workshops and courses we have been using the LilyPad to engage middle and high-school students (ages 11-18) in computing and electronics. In each course students learn basic circuitry and programming and then design and construct an interactive garment that is demoed to friends and family at an exhibition/fashion show. Figure 6 shows images from a few of these sessions.

Figure 4. Images from Electronic Fashion workshops. Left: two students work on their designs. Center: a young woman models the e-textile she built, a sweatshirt with electroluminescent wire and LEDs. Right: two teenagers have fun with a touch sensitive shirt. The shirt, built by the young woman in the picture, makes sounds when someone squeezes her waist.

One of the most interesting outcomes from these experiences was our ability to attract voluntary participation from large numbers of young women, who—once they arrived in workshops—adopted engineering skills with gusto to complete functional and sophisticated designs (Buechley, 2008).

Margolis and Fisher's (2001) groundbreaking study on gender in computer science focused on "Unlocking the Clubhouse". In their report on the study they illustrate how traditional computing culture functions as a white/Asian boys' club and argue that it is crucially important to unlock this clubhouse to make it more accessible to women and minorities. Our experiences suggest a different approach, one we call "Building New Clubhouses". Instead of trying to fit people into existing computing cultures, we want to spark and support new ones. Rather than trying to recruit young women to robotics clubs and classes, we engage them in computation through electronic textile clubs and classes—venues that young women flock to with no prompting.

We believe that cultural factors, more than a lack of aptitude or intrinsic interest, make computer science inaccessible and unappealing to many students. By making computation more accessible and building computers that look and feel different from traditional ones—computers that are fuzzy, colorful, and feminine, for example—we can begin to change and broaden the culture of computation. We can begin to get a diverse range of people excited by the ways that computers can be used to build beautiful, expressive, and useful objects that are different from anything that has been built in the past.

Since the introduction of the LilyPad Arduino, a community of educational technology researchers has begun to adopt our tools and employ them in similar settings (Katterfeldt, 2009; Ngai, 2009) and we are optimistic that new real-world cultures are beginning to flourish outside of our research lab.

Connecting the science classroom and tangible interfaces: the Bifocal Modeling framework

Paulo Blikstein

Fifteen years ago, few would have predicted that children would be doing advanced robotics in middle-school. Indeed, since the seminal work by Papert, Martin, and Resnick (Martin, 1993; Resnick, et al., 1998; Resnick, Ocko, and Papert, 1991), the launch of the Lego Mindstorms platform, and the appearance of robotics competitions across the country, robotics has become a common activity in public and private schools. However, the learning revolution predicted by its proponents is still far away – such activities are oftentimes attended by males, too focused on competitions and prescribed, standardized "challenges," and disconnected from the school curriculum. In most schools, robotics teachers conduct activities regardless of what happens in the science or math classroom.

At the same time, science classrooms and laboratories are not well suited to support students for authentic scientific inquiry, developing and investigating their own scientific hypothesis. For example, a student examining an acid-base reaction in a laboratory might identify the chemical elements involved and even hypothesize as to their proportions and concentrations, but the investigation cannot dive deep into the chemical mechanisms. Later, in the classroom, he will learn about chemical equations and theories which bear little resemblance, in terms of scale and mechanism, to the phenomenon observed in the laboratory. Bifocal Modeling (Blikstein & Wilensky, 2006) is a framework to link these disconnected types of activities and environments (robotics and computational manipulatives, science laboratories, and theoretical content in science), providing continuity between observation, physical construction of artifacts, and model-building. As this modeling platform enables seamless integration of the theoretical/computational models and the physical world, allowing modelers to focus simultaneously on their 'on-' and 'off-screen' models, I termed it *bifocal modeling*.

When building a bifocal model, students have three main tasks. First, they build a computer model of the phenomenon using various computer modeling platforms (in particular, I use NetLogo (Wilensky, 1999) in my studies). This model should encapsulate students' hypotheses about a given scientific phenomenon. Second, students use electronic sensors and low-cost analog-to-digital interfaces, such as the GoGo Board (Sipitakiat, Blikstein, & Cavallo, 2004), to build their own sensor-equipped "science lab." Finally, students run both models connected in real-time as to validate, refine, and debug their hypotheses using real-world data. The computer screen becomes a display for the computer model, which is a proceduralization, through programming, of equations, text, or other representations of scientific content, and the actual phenomenon, which is discretized by means of sensors and other laboratory apparatus (see Figure 5, for a model investigating heat transfer in a copper wire, and Figure 6, for a model of acid-base reactions). Because the computer models are carefully constructed to imitate the phenomenon's visual language, the bifocal methodology minimizes interpretive challenges. That is, the seen and the hypothesized are displayed such that their perceptual differences are backgrounded and, therefore, their procedural differences are more likely to be revealed. By thus utilizing the power of computation and representation, bifocal modeling constitutes a multi-disciplinary research tool that offloads aspects of both the interpretive and menial burden of scientific practice, freeing cognitive, discursive, and material resources that can thus be allocated toward validation of the hypotheses.

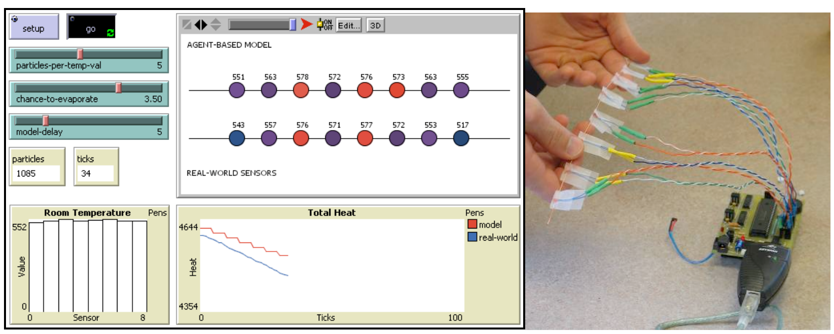

Figure 5. A computer model of heat transfer, with the side-by-side visualization (left), and the physical model (right), with a copper wire hooked to eight temperature sensors.

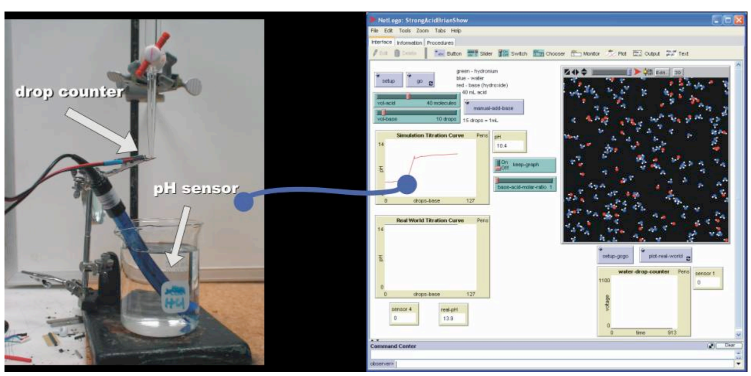

Figure 6. A model of an acid-base reaction, with the physical apparatus (left) and the NetLogo computer model (right), side-by-side.

In particular, in previous pilot studies (Blikstein & Wilensky, 2006), students who built bifocal models attended to phenomenal factors which were not mentioned by students in a second group who did on-screen-only models, such as energy loss, reversibility, synchronicity, and precision. The student who built the heat transfer model in Figure 5, for example, wanted to test how different metals would behave when heated. Coming in to the project, he had two hypotheses about the nature of each of the foci of bifocal modeling. He supposed that it should be relatively straightforward to build: (a) an artifact that enables the measurement of the target phenomenon; and (b) a computer-based procedure that emulates this phenomenon. Both hypotheses proved incorrect. The unsettling element in his model, which triggered the frustration of his expectations, was *time*. Upon completing the physical model and connecting it to the computer model, he realized that there was a fundamental (and hard) problem to be addressed: synchronicity. Sensors were sending temperature data twenty or thirty times a second, but the computer was calculating new temperatures for the virtual agents several thousands of times a second. Which "side" should be in control? Should the computer model be slowed down to match the real-world data, or should the sensor data be manipulated by software to fit into the timing scheme of the computer model? Both options have significant implications for modeling, and speak to the modeling endeavor itself. If the computer timing would prevail, the sensor data would be greatly 'stretched', and perhaps become meaningless. In the physical model, the inch that separated two temperature sensors contained billions of atoms. In the computer model, that same distance contained just a couple of computational agents. The nanosecond events taking place in the real material would have to be somehow converted to the model scale.

The student spent a significant part of the workshop thinking about this issue and what was, in fact, the objective phenomenon being modeled. Was it 'what happens when you heat a wire' or is it 'the concept of heat flow?' In traditional textbooks, chapter titles disclose 'what is to be learned,' such that learning is concept-driven, whereas his experience was phenomenon-driven (see Papert, 1996, on the 'project-before-problem' principle). As the seen and the hypothesized are displayed simultaneously, their perceptual differences are backgrounded and, therefore, crucial procedural differences can be revealed and problematized.

This is one of the many case studies documenting students' experience building bifocal models. First, by connecting science content and construction of artifacts, I allow students to better transition between what is learned in the classroom and what they build with technology. Second, the construction of bifocal models, by making students connect computer models and physical sensors in real-time, introduced novel, deep issues that speak to the nature of science and the process of modeling, namely, friction/energy loss, precision, scale, time, coefficients, scale conversion, and synchronicity. Third, the motivation and engagement that is commonly observed in "hands-on" technology-rich building activities is mobilized towards creating content-driven connections with the learning of science and mathematics.

Tangible Programming in Formal and Informal Educational Environments

Michael S. Horn

Real-world learning environments are complex and often chaotic places. Teachers in classrooms must learn to balance the needs of anywhere from 15 to 30 students at a time with the demands of curriculum and the constraints of a regimented school day. In non-school environments such as science museums, the challenge is different. Program developers and exhibit designers must work without the structure and guidance provided by teachers and curriculum, devising activities and exhibits that engage a diverse audience and promote self-guided learning. For educators, the decision to incorporate computational learning activities in these setting can be fraught with risk (AAUW, 2000; Cuban, 2001). Teachers may feel a sense of loss of control and self-doubt about their own proficiency with technology (AAUW, 2000), and desktop computers, designed primarily as single-user productivity tools for businesses, can be less than ideal for many educational applications (Scott, Mandryk, & Inkpen, 2003). Likewise, in museums, although computer-based exhibits can be very engaging for individual visitors, they are often detrimental to the interactions of social groups as a whole (e.g. Hornecker & Stifter, 2006). For the past four years I have been exploring the potential of tangible interaction to address these issues. Here I briefly describe some of this work in both formal and informal educational settings. I conclude with a brief argument for a focus on creating *hybrid tangible interfaces* that combine tangible and graphical interaction into a single system, thus giving users the freedom to select an input modality to meet their current needs or preferences.

Figure 7. Tern allows children to program by connecting interlocking wooden blocks.

My research has involved a computer programming language called *Tern* (Figure 1), a tangible interface designed for children to control robotic creations. Rather than program with a mouse or keyboard, children use a collection of interlocking wooden blocks to create physical algorithmic structures. These blocks are compiled into digital code using low-cost computer vision techniques. With tradition programming languages, children are involved in the creation digital artifacts. One goal of the Tern project is to transform these digital artifacts into physical artifacts—highly visible products of student work and can become part of presentations and discussions in learning environments.

Tangible Programming in Science Museums

In 2008 I worked with colleagues at Tufts University to evaluate the use of Tern as part of a computer programming and robotics exhibit at the Boston Museum of Science (Horn, Solovey, Crouser, & Jacob, 2009). For this study, we created two interaction conditions: a graphical condition that presented museum visitors with a computer mouse and a display, and a tangible condition that presented museum visitors with a collection of wooden programming blocks. We then observed museum visitors as they interacted with the exhibit using one condition or the other on alternating weekend days (e.g. tangible on Saturday and graphical on Sunday). These observations revealed certain advantages for the tangible programming interface from the standpoint of informal science education. In particular, the tangible blocks were more inviting to visitors, and they were better at facilitating active collaboration. These findings were especially strong for children and for girls in particular. For example, roughly 33% of girls who noticed the mouse-based version of the exhibit stopped to try it. This number rose to 85% of girls in the tangible condition. For other measures there were no significant differences between conditions. This included amount of time spent interacting with the exhibit, the number of programs created, and the length and complexity of those programs.

Tangible Programming in Kindergarten

I have also been involved in research investigating the use of Tern in classrooms as part of the Tangible Kindergarten project at Tufts University. The goals of this project include (a) creating in-depth computer programming and robotics curriculum for use in kindergarten classrooms; (b) creating age-appropriate programming technology; and, (c) developing a richer understanding of young children's ability to engage powerful ideas from computer programming and robotics. As part of this project, we piloted an eight-hour curriculum with four classrooms of kindergarten children (ages 5–6) at a local elementary school. We divided these classrooms into two conditions, tangible and graphical. In the graphical classrooms, children used four desktop computers, while in the tangible classrooms, children created programs with wooden blocks.

Based on observation notes and an analysis of videotape, we found that children were able to easily manipulate the tangible blocks to form their own programs. For children in the graphical condition, however, we observed a range of capabilities in terms of being able to manipulate the computer mouse. We also found that the students were able to differentiate the blocks and discern their meanings in both the graphical and the tangible conditions. In terms of the curriculum, for certain activities and for certain children, the tangible version of Tern was clearly advantageous. For example, children could participate in whole-class discussion in a hands-on way with the tangible blocks. On the other hand, some of the most independent group work that we observed was with the graphical interface. Overall, the results of this study were mixed. Some children demonstrated surprisingly sophisticated understandings of computer programming and robotics concepts, while other children struggled throughout the curriculum unit. Since this study, the Tangible Kindergarten team has done much to improve both the curriculum and the programming technology; however, there is much to be done to refine our understanding of children's ability to participate in these types of activities in meaningful ways.

Argument for a Hybrid Approach

I conclude with a proposal that it is advantageous to combine tangible interaction with more traditional interfaces to create hybrid systems. This approach leads to several immediate advantages. The most important is that it gives actual participants in learning environments—teachers, students, museum staff, etc.—the flexibility to select the type of interaction most appropriate for a given learning situation. This flexibility is especially important in classroom settings, where teachers often determine if and when a particular technology will be used (Cuban, 2001). In addition, the use of hybrid interfaces means that not every feature of a software system need be implemented tangibly. For example, saving student work on a file system might be a feature better left to the graphical version of the interface. Beyond these immediate advantages is the potential to provide layered scaffolding as students progress toward increasingly authentic programming environments. For example, students might start with a tangible system and then transition to a graphical system with increased capabilities and complexity.

References

AAUW Educational Foundation. (2000). Tech Savvy: Educating Girls in the New Computer Age. American Association for University Women.

Blikstein, P., & Wilensky, U. (2006). *The Missing Link: A Case Study of Sensing-and-Modeling Toolkits for Constructionist Scientific Investigation.* Paper presented at the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, The Netherlands.

Bruner, J. (2004) Towards a Theory of Instruction. Belknap Press.

Buechley, L., Eisenberg, M., Catchen, J., and Crockett, A., The LilyPad Arduino: Using Computational Textiles to Investigate Engagement, Aesthetics, and Diversity in Computer Science Education, In Proceedings of the SIGCHI conference (CHI 2008), Florence, Italy, April 2008, pp. 423-432.

Cole, M., and Cole, S. (2001) The Development of Children, Fourth Edition. New York, NY : Worth Publishers.

Cuban, L. (2001). Oversold and Underused: Computers in the Classroom. Cambridge, MA: Harvard Univ. Press.

Frei, Phil. (2000). curlybot: Designing a new class of computational toys. Master's thesis, Massachusetts Institute of Technology.

Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J.K. (2009). Comparing the use of tangible and graphical programming interfaces for informal science education. In Proc. ACM Conference on Human Factors in Computing Systems CHI'09, 975-984.

Hornecker, E., & Stifter, M. (2006). Learning from interactive museum installations about interaction design for public settings. In Proc. Australian Computer-Human Interaction Conference OZCHI'06.

Katterfeldt, E., Dittert, N., and Schelhowe, H. 2009. EduWear: smart textiles as ways of relating computing technology to everyday life. In Proceedings of the 8th international Conference on interaction Design and Children (IDC 2009), Como, Italy, June 2009, pp. 9-17.

Margolis, J. and Fisher, A. [2001] Unlocking the clubhouse. MIT Press, Cambridge, MA.

Martin, F., Resnick, M. (1993). Lego/Logo and electronic bricks: Creating a scienceland for children. In D. L. Ferguson (Ed.), *Advanced educational technologies for mathematics and science*. Berlin, Heidelberg: Springer-Verlag.

Ngai, G., Chan, S. C., Cheung, J. C., and Lau, W. W. 2009. The TeeBoard: an education-friendly construction platform for e-textiles and wearable computing. In Proceedings of the 27th international Conference on Human Factors in Computing Systems (CHI 2009). Boston, MA, USA, April 2009, pp. 249-258.

Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas.* New York: Basic Books.

Papert, S. (1996). An Exploration in the Space of Mathematics Educations. *International Journal of Computers for Mathematical Learning, 1*(1), 138-142.

Piaget, J. (1976). The grasp of consciousness. Cambridge: Harvard University Press.

Raffle, Hayes (2008). Sculpting Behavior: A Taigible Language for Hands on Play and Learning. PhD Thesis, Massachusetts Institute of Technology.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., et al. (1998). *Digital manipulatives: new toys to think with.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Resnick, M., Ocko, S., Papert, S. (1991). Lego/Logo: Learning through and about design. In I. Harel, Papert, S. (Ed.), *Constructionism*. Norwood, NJ: Ablex.

Scott, S., Mandryk, K., & Inkpen, K. (2003). Understanding children's collaborative interactions in shared environments. Journal of Computer Assisted Learning, 19, 220–228.

Sipitakiat, A., Blikstein, P., & Cavallo, D. P. (2004). *GoGo Board: Augmenting Programmable Bricks for Economically Challenged Audiences.* Paper presented at the International Conference of the Learning Sciences, Los Angeles, USA.

Wilensky, U. (1999). NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling. http://ccl.northwestern.edu/netlogo.

Acknowledgments

Michael Horn: Many colleagues at Tufts University and the Boston Museum of Science have contributed to this research including Marina Bers, Robert Jacob, Jordan Crouser, Erin Solovey, Taleen Agulian, and Dan Noren.

Paulo Blikstein: Uri Wilensky at Northwestern University, who advised this work, and many colleagues at the Center for Connected Learning and Computer-Based Modeling, as well as Dor Abrahansom at UC Berkeley.

Leah Buechley: Mike Eisenberg, Nwanua Elumeze, Ann Eisenberg, and colleagues at the University of Colorado, Mark Gross, and colleagues at the MIT Media Lab.