Start 'em Young: Digital Music Instruments for Education

Jiffer Harriman
ATLAS Research Institute
University of Colorado
1125 18th St. #223
Boulder, Colorado 80309
harrimj@colorado.edu

ABSTRACT
Designing and building *Digital Music Instruments* (DMIs) is a promising context to engage children in technology design with parallels to hands on and project based learning educational approaches. Looking at tools and approaches used in STEM education we find much in common with the tools and approaches used in the creation of DMIs as well as opportunities for future development, in particular the use of scaffolded software and hardware toolkits. Current approaches to teaching and designing DMIs within the community suggest fruitful ideas for engaging novices in authentic design activities. Hardware toolkits and programming approaches are considered to identify productive approaches to teach technology design through building DMIs.

Author Keywords
NIME, education, STEM, toolkits, digital music instruments

ACM Classification
A.1 [Introductory and Survey] K.3.1 [Computers and Education] Computer Uses in Education

1. Introduction
Designing and building *Digital Music Instruments* (DMIs) is an interdisciplinary endeavor well suited to engage children in hands on Science, Technology, Engineering and Math (STEM) learning. Building DMIs provides opportunities to engage learners with electronics, programming and interaction design through the creative context of music. Popular activities used for STEM education such as robotics, game design and e-textiles combine design and engineering. The premise of this paper is that *building DMIs is an understudied context for hands-on STEM education through personally motivated design projects*.

I first overview contexts and approaches common in STEM education. Followed by an overview of existing approaches found in teaching DMI design and design tools that suggest potential starting points for this space.

2. LEARNING BY DOING
Studies of constructionist learning show the efficacy of environments such as "computer clubhouses" [12], a precursor to today's public maker spaces where young people engage with programming, electronics, and digital fabrication. These activities [9] have become integral to after school programs and, increasingly, in the classroom. It is easy to see why. Imagining, designing and creating a technologically rich artifact engages students in engineering design in meaningful ways that meet the challenges of the consensus document *A Framework for K-12 Science Education*. [20] Resnick and Rosenbaum describe the value of tinkering as an act closely resembling the engineering design process noting that "[w]hen people are tinkering, they are constantly trying out ideas, making adjustments and refinements, then experimenting with new possibilities, over and over and over." [25]

Turkle and Papert advocate for a *bricolage* programming style (a reflexive approach with parallels to tinkering) as a means to make computer science appealing to a wider audience. [31] McLean detailed how bricolage programming is used in the arts involving tools common to building DMIs including Pd, Max, ChucK and SuperCollider. [16] He describes this way of working as a creative feedback loop that has much in common with the creative spiral described by Resnick in creating tools to enable children to design with electronics and programming (figure 1). [23] This synergy of working styles highlights the opportunity to use the design of DMIs as a viable approach to engaging children in technology design.

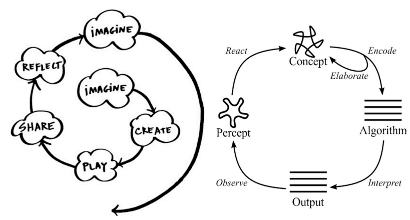

Figure 1: Creative Thinking Spiral (left); Process of action and reaction in bricolage (right)

2.1 Tools and Scaffolding
An important feature of educational tools and environments is scaffolding, a "knowledgeable other" that guides students through tasks they would otherwise be unable to complete on their own. [21] Scaffolding, which could be a mentor or a tool that guides the student, is common in children's programming languages such as Scratch [24] and Etoys [13], which prevent illegal coding constructs. LittleBits is a hardware example that uses magnetic connectors, which prevent wiring incorrect circuits. [1] The discussion in the future work section is predominantly based on this concept.

3. CONTEXTS
Contexts used to engage children in STEM learning have parallels to leisure and fun activities. Examples discussed here show how game design, storytelling, videos, art projects and music have been used to teach programming and electronics to children.

3.1 Programming
A child's first encounter with computers is often through play. Inviting children to become designers of educational games, Kafai [11] shows how they build deeper connections with underlying learning goals. Similarly, Webb et al [34] use a guided discovery based approach to teach computer science principles through game design. This approach of *leveraging personalization to increase ownership and motivation* is an obvious fit with designing DMIs if we let students *define their interaction paradigm, physical layout, associated sound and even compositional components.*

Dann et al [6] describe how using the Alice programming language to control animations in a virtual 3D environment helps learners comprehend concepts about the execution and state of a program. Kelleher et al [14] found that students who used a version of Alice with supports for *storytelling* spent more time programming and were more interested in working on the programming activity again. The use of storytelling parallels Guzdial's media computation approach to teaching computer science to non-majors where students write programs to manipulate sound, images and movies — *computation for communication* [8].

Students need a reason to take up creative expression by programming: a story to tell, a game to create, a song to sing. To remain engaged they need to see how their work directly affects a 3D scene, an animation, or a melody.

3.2 Hands-on Hardware

In holding workshops to introduce children to electronics through robotics Rusk et al [26] found that different students are attracted to different types of robotic activities, *hence the importance of multiple pathways of engagement*. They describe four strategies which *naturally map to musical endeavors*: (1) focusing on themes, not challenges; (2) combining art and engineering; (3) encouraging storytelling; (4) organizing exhibitions rather than competitions. Resnick [23] describes the open ended nature of the MIT Crickets noting that the target application of Crickets is in creative and artistic projects, which tend to attract more diverse participants, in contrast to traditional robotics.

Blikstein and Sipitakiat [2] argue for critically considering the affordances of various microcontroller designs for children to enable an appropriate level of transparency for the audience and desired learning outcomes. In teaching robotics to middle/high schoolers and undergrads they identify challenges encountered by students including difficulty understanding the difference between analog, PWM and digital pins, the need for pull-up resistors and the architecture of a solderless breadboard, reporting that students use them without understanding the rationale for the physical connections they make. Qiu, et al [19] suggested a solution to these challenges introducing a curriculum for teaching through building and programing computational textiles. Their curriculum makes use of scaffolding using the ModKit language and LilyPad Protosnap board, high-level design abstractions that help novice learners avoid getting stuck on programming and circuit details.

Building DMIs provides opportunities to engage with hardware in a compelling and rich design task. However existing educational tools limit musical expressiveness as they are not designed explicitly for music while programming languages for music are not designed for children.

3.3 Music in Educational Tools

Music offers a context that appeals to a broad population and enables an equally broad spectrum of personalized projects. To capitalize on this, Scratch has built in objects to create music (figure 2). This simple set of controls includes playing a sample, a note from a list of instruments, or a percussion sound, as well as changing the tempo and volume. These provide enough flexibility to program a wide variety of projects. Searching the Scratch website for projects with the keyword "music" returns an astonishing 6 million entries[1] demonstrating the demand for music in children's construction/educational tools. Another key to the Scratch environment is reuse, making it easy to start from someone else's program, remix it and make it your own.

Drawdio [28] and Makey Makey [29] promote exploration of novel sound control through the physical world. While these projects certainly capture the imagination they only scratch the surface of the richness and potential available in designing DMIs.

There are opportunities to make these experiences more engaging by enabling more expressivity and creativity. How can we create intuitive and fun to use designs that can enable richer explorations?

Figure 2: Scratch sound objects

3.4 Music as a Motivator

Some have used music and the creation of DMIs to engage children in programming, electronics and design. Using high level GUI abstractions to engage primary school children in instrument design Trappe [30] identifies "musical playfulness" as a key to success concluding, music controller construction is a context that nurtures self-motivated creation, exploration and play. Sawyer et al [27] describe a workshop where children use hardware sensors and a graphical programming environment to create musical instruments noting evidence of connections to design processes. A related study by Bukvic et al [3] describes the use of "granularity" as a means to provide multiple points of entry, and in enabling an adaptive tool (Pd-L2Ork, also used by Sawyer) that can match the educational model and skill level of the audience.

These projects establish music and the design of DMIs as a promising avenue for teaching programming, electronics and interaction design. However further research is needed to clarify: What specifically can be learned in these settings? What is an appropriate level of scaffolding for different age groups to best facilitate learning? And what pedagogical approaches should be used?

4. TEACHING DMI DESIGN

In considering the design of DMIs as a context to engage children in programming and electronics design it is relevant to look at teaching approaches as well as design considerations and the tools used.

4.1 Designing and Evaluating DMIs

In [33] Verplank et al describe one of the first NIME based courses taught at CCRMA to teach controller development with the theme of *buttons* and *handles*. They describe how students develop practical skills while creating new interfaces with a variety of novel designs concluding, "the direct engagement in an expressive realm like music can generalize to a wide range of human-machine controllers". Malloch et al identify the defining feature of DMIs as the separation of the human interface and sound production. [15] In designing a DMI the primary task is mapping this relationship. Wessel [35] discusses metaphor as a powerful approach to creating expressive gesture-to-sound-mappings. He suggests novel approaches to sound control, including "tonal pitch space" and "timbre space" as well as metaphors for creating music controllers: *drag and drop*, *scrubbing*, *dipping*, and *catch and throw*. The concept of metaphor is also a component of Verplank's [32] interaction design framework, where he promotes idea sketching as a crucial tool for interaction designers that combines learning by doing with anticipation and reflection. Verplank expands Wessel's task of mapping gesture to sound to include the way a user "knows" and the way an interaction "feels".

[1] scratch.mit.edu accessed 1/30/2015

In [4] and [5] Cook offers guiding principles to consider in designing new instruments. Some of his principles may only apply to "serious" NIME practitioners but several are highly relevant for introducing NIMEs to novices as he points out *Music+Science is a great teaching/marketing tool* and *The younger the student the more fearless*. His idea to *Make a piece, not an instrument* puts the designer in the composer's shoes, an opportunity for adding a personal layer to projects. *Smart instruments are often not smart* and *Instant music subtlety later* match well with a beginner's mindset, and the creation of accessible, easy to understand instruments for novices. The suggestion *Everyday objects suggest amusing controllers* gets at the playful potential of designing DMIs and is likely to be attractive to children.

Jordà and Mealla introduced a method for teaching DMI design and evaluating the results with a focus on self-reflection and an iterative process of refinement [10]. The course they describe focuses on the common NIME challenges of *expressiveness* and *mapping*. In asking the participants to reflect on their design by watching recordings of their own performances they put the designers in the audience perspective as a novel way to capture the variety of viewpoints [17]. They found when evaluating their DMIs, students with previous music experience had a better grasp of the concepts of expressiveness and virtuosity. This suggests a potential advantage in teaching nuanced concepts of interaction design to students with musical training.

5. TOOLS FOR BUILDING DMIs

In describing a class at CCRMA, Wilson et al answer their own question of "Why Microcontrollers?" with *pedagogy*, arguing it provides the opportunity for students to learn about things such as programming, digital logic, and A/D conversion [36]. The tools of design directly impact what can and sometimes, must be learned to use them. A toolkit such as I-CubeX[2] configured to generate MIDI messages can be used to directly control an off-the-shelf software instrument without any programming or circuit building required. However, such a system still holds potential to explore concepts like *expressiveness* and *mapping* while exposing new designers to sensor technologies.

The challenges identified by Blikstein and Sipitakiat [2] conspire to intimidate a student rather than build confidence. Instead, it is worth considering toolkits that lower the barrier to entry of designing with electronics for the creation of DMIs, especially for young audiences.

A variety of toolkits for solderless sensor interfacing exist with varying levels of learning curve, openness and expandability. Following is a review of currently available toolkits with potential as DMI construction kits. There are others, but the kits here represent the most common platforms. They are organized into three categories, *general purpose*, *education* and *music* toolkits.

5.1.1.1 General Purpose Toolkits

These toolkits were created to enable fast prototyping and provide the most flexibility providing software libraries to accompany their hardware modules and support serial communication. General purpose toolkits include Phidgets [7] Grove[3], Tinkerkit[4], and .Net Gadgeteer[5]. Phidgets is the onlyof these that is not open-source. .Net Gadgeteer programming is in C# with no support for Arduino. For input hungry jobs (like many instruments) the Gadgeteer has the most built in I/O but these all support I2C for expansion options. In [18] Overholt discussed the CUI32Stem with the Grove sensor kit as a platform for rapid prototyping of DMIs.

5.1.2 Toolkits for Education

In [22] Resnick introduced the "programmable brick" a precursor to the PicoBoard, suggesting building a Lego musical instrument as a potential project. Today Scratch sound objects (figure 2) can be controlled with the PicoBoard[6] or Blikstein's GoGo Board [2] to enable children to build. However the sound programming in Scratch limits *mapping* possibilities and a meaningful exploration of *expressiveness*.

The GoGo Board and Hummingbird Kit[7] can be programmed with Arduino or with programming languages made for education. The GoGo Board documents its serial protocol which could be used for communication with audio programming languages.

5.1.3 Toolkits for Music

Commercial products for making custom interfaces offer more flexibility in terms of I/O and software interface, although none are open-source. I-CubeX offers applications to route MIDI messages as well as C++ APIs to communicate directly with the hardware with 32 analog inputs. Livid Instruments' Builder Kits[8] support as many as 64 analog inputs and 128 digital inputs and will register as a USB MIDI device.

6. Discussion

Toolkits lower the barrier to entry and make it possible for children to design with programming and electronics. However the affordances of a toolkit should be considered with learning goals. Additionally we should consider how we might create pathways from heavily scaffolded approaches to more open ended tools commonly used in the creation of DMIs thus creating a migration path for novices.

6.1 Areas for Future Work

Literature on teaching children technology design suggests scaffolding is a productive way to engage novices with otherwise challenging tasks. But how should hardware toolkits be utilized in teaching DMI design? What software tools are needed? And what concepts of programming, electronics, interaction design, music and more can be taught through DMI design? Further research is needed to answer these questions and to begin developing targeted tools suited for the task.

In developing tools for educational DMI design we should consider the learning goals, options for scaffolding, teaching approaches as well as practical considerations like cost and the sensors used to enable understandable but varied and expressive interfaces. An understanding of what has been successful in other related fields of robotics and game design is informative but there is no silver bullet. New approaches need to be field tested with children and other novices to build knowledge of promising tool and pedagogical designs. Is it possible to create tools for DMI design which can be easily understood by any motivated student or teacher?

Such tools could be used to enable children to intuitively map gesture to control sound enabling an engaging context for teaching the next generation of interaction designers through the engaging context of DMI design.

7. Summary

Designing and building DMIs is an understudied context for STEM education that can be used to engage children in

[2] http://infusionsystems.com/
[3] http://www.seeedstudio.com/wiki/GROVE_System
[4] http://store.arduino.cc/category/16
[5] http://gadgeteer.codeplex.com/
[6] http://wiki.scratch.mit.edu/wiki/PicoBoard
[7] http://www.hummingbirdkit.com/
[8] http://shop.lividinstruments.com/builder-diy/

personally motivated projects. The primary task of *mapping* gesture to sound to create *expressive* interfaces and the use of *metaphors* make building DMIs a rich context for interdisciplinary learning. Existing sensor toolkits offer starting points for engaging children with hardware. The limited research into building DMIs with children shows promise although questions remain about tools and approaches. These suggest areas for future research including the development of hardware and software tools and accompanying pedagogy for teaching DMI design to children to further explore the opportunities for learning.

8. REFERENCES

[1] Bdeir, A. (2009). Electronics as material: littleBits. Proceedings of the 3rd International Conference on Tangible Embedded, Embodied Interaction, 397–400.

[2] Blikstein, P., & Sipitakiat, A. (2011). "QWERTY and the art of designing microcontrollers for children." Proceedings of the 10th International Conference on Interaction Design and Children - IDC '11, 234–237.

[3] Bukvic, I., Baum, L, Layman, B., & Woodard, K. (2011). "Granular Learning Objects for Instrument Design and Collaborative Performance in K-12 Education", 12–14.

[4] Cook, P. (2001). Principles for designing computer music controllers. Proceedings of the 2001 Conference on New Interfaces for Musical Expression, 1–4.

[5] Cook, P. (2009). Re-designing principles for computer music controllers: A case study of SqueezeVox Maggie. Proceedings of the 9th Conference on New Interfaces for Musical Expression, 11, 218–221.

[6] Dann, W., Cooper, S. and Pausch, R.. "Making the connection: programming with animated small world." ACM SIGCSE Bulletin. Vol. 32. No. 3. , 2000.

[7] Greenberg, S., & Fitchett, C. (2001). Phidgets: easy development of physical interfaces through physical widgets. Proceedings of the 14th Annual ACM Symposium on User Interface Software Technology, 3(2).

[8] Guzdial, M. (2004). Computers for Communication , Not Calculation : Media as a Motivation and Context for Learning Andrea Forte, 00(C), 1–10.

[9] Honey, M. and Kanter, D.E. (eds.) Design | Make | Play: Growing the Next Generation of STEM Innovators, Routledge 2013.

[10] Jordà, S., & Mealla, S. (2014). A Methodological Framework for Teaching, Evaluating and Informing NIME Design with a Focus on Expressiveness and Mapping. Proceedings of the 15th Conference on New Interfaces for Musical Expression, 233–238.

[11] Kafai, Yasmin B. Minds in play: Computer game design as a context for children's learning. Routledge, 1995.

[12] Kafai, Yasmin B., Peppler, K. and Chapman, R. (2009). The Computer Clubhouse: Constructionism and Creativity in Youth Communities. Technology, Education--Connections. Teachers College Press, New York

[13] Kay, Alan. "Squeak etoys, children & learning." online article 2006 (2005).

[14] Kelleher, C., Pausch, R., & Kiesler, S. (2007). "Storytelling Alice motivates middle school girls to learn computer programming". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07, 1455.

[15] Malloch, J., Sinclair, S., Hollinger, A., & Wanderley, M. M. (2011). Input Devices and Music Interaction, (Idmil), 67–83.

[16] Mclean, A., & Wiggins, G. (2010). Bricolage Programming in the Creative Arts.

[17] O'Modhrain, S. (2011). A framework for the evaluation of digital musical instruments. Computer Music Journal, 28–43.

[18] Overholt, D. (2012). Musical Interaction Design with the CUI32Stem: Wireless Options and the GROVE system for prototyping new interfaces. *Proceedings of 12th Conference on New Interfaces for Musical Expression*. (2012)

[19] Qiu, K., Buechley, L., Baafi, E., & Dubow, W. (2013). "A curriculum for teaching computer science through computational textiles". Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13

[20] Quinn, H., & Bell, P. (2013). How designing, making, and playing relate to the learning goals of K-12 science education. In M. Honey & D. Kanter (Eds.), Design, make, play: Growing the next generation of STEM innovators (pp. 17-33). New York: Routledge.

[21] Reiser, B, Tabak, I. "Scaffolding." The Cambridge handbook of the learning sciences. Second Edition (in press).

[22] Resnick, M. (1993). Behavior construction kits. Communications of the ACM.

[23] Resnick, M. (2009). Sowing the Seeds for a more Creative Society. Proceedings of the 27th International Conference on Human Factors in Computing Systems - CHI 09, 30.

[24] Resnick, M., & Maloney, J. (2009). Scratch: programming for all. Communications of the ACM

[25] Resnick, M. & Rosenbaum, E. (2013). Designing for tinkerability. In M. Honey & D. Kanter (Eds.), Design, make, play: Growing the next generation of STEM innovators (pp. 163- 181). New York: Routledge.

[26] Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2007). "New Pathways into Robotics: Strategies for Broadening Participation." Journal of Science Education and Technology, 17(1), 59–69. doi:10.1007/s10956-007-9082-2

[27] Sawyer, B., Forsyth, J., O'Connor, T., Bortz, B., Finn, T., Baum, L., Bukvic, I., Knapp, B., Webster, D. (2013). Form, Function and Performances in a Musical Instrument MAKErs Camp, Proceedings of the 44th ACM technical symposium on Computer science education. 669–674.

[28] Silver, Jay. "Awakening to maker methodology: the metamorphosis of a curious caterpillar." *Proceedings of the 8th International Conference on Interaction Design and Children*. ACM, 2009.

[29] Silver, J., & Shaw, D. (2012). Makey Makey: improvising tangible and nature-based user interfaces. Proceedings of the Sixth International Conference on Tangible Embedded, Embodied Interaction, 367–370.

[30] Trappe, C. (2012). "Making Sound Synthesis Accessible to Children." Proceedings of the 2012 conference on New Interfaces for Musical Expression (NIME).

[31] Turkle, S., & Papert, S. (1991). Epistemological pluralism and the revaluation of the concrete. Constructionism.

[32] Verplank, W. 2009. Interaction Design Sketch Book. Unpublished paper. Available on-line at www.billverplank.com/IxDSketchBook.pdf

[33] Verplank, B., Sapp, C., & Mathews, M. (2001). A course on controllers. Proceedings of the 2001 Conference on New Interfaces for Musical Expression.

[34] Webb, D. C., Repenning, A., & Koh, K. H. (2012). Toward an emergent theory of broadening participation in computer science education. Proceedings of the 43rd ACM Technical Symposium on Computer Science Education

[35] Wessel, D., & Wright, M. (2002). Problems and Prospects for Intimate Musical Control of Computers. Computer Music Journal, 26(3), 11–22. doi:10.1162/014892602320582945

[36] Wilson, S., Gurevich, M., Verplank, B., & Stang, P. (2003). Microcontrollers in music HCI instruction: reflections on our switch to the Atmel AVR platform. Proceedings of the 2003 conference on New Interfaces for Musical Expression, 24–29.