The Development and Use of

Scaffolded Design Tools for Interactive Music

by

Jeffrey Wood Harriman, Jr.

B.S. University of Colorado, 2002

M.A. Stanford University, 2012

A thesis submitted to the Faculty

of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

ATLAS Institute, College of Engineering

2016

This Thesis entitled:

Scaffolded Design Tools for Interactive Music

written by Jeffrey Wood Harriman, Jr.

has been approved for the College of Engineering

Michael Theodore

Mark D. Gross

Mike Eisenberg

Tom Yeh

Miller S. Puckette

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Harriman, Jeffrey Wood, Jr. (Ph.D., Technology, Media, and Society)

The Development and Use of Scaffolded Design Tools for Interactive Music

Thesis directed by Dr. Michael Theodore

In music, and the arts, electronic and computational technologies have created new ways of working, enabling new forms of creative expression and new experiences. New media artists and experimental musicians are exploring the possibilities enabled by computational, interactive and robotic technologies, and digital fabrication tools to enable new forms of expression and to create new experiences for audiences. The artists and musicians who employ these technologies at the bleeding edge create new techniques, hardware and software. New mediums and approaches like augmented reality, 3D printing, interactive and kinetic sculpture, new interfaces for musical expression, and robotic musical instruments are nascent compared to the violin or oil paints. These mediums represent new opportunities for that require new tools and approaches to realize their potential. This dissertation is about creating and using such design tools for interactive music.

Dedication

This dissertation is dedicated to my wife Jamie. I couldn’t have made it here without your loving support. And to my parents, for encouraging curiosity.

Acknowledgements

I am incredibly grateful for the past 6 years of graduate school and all the people who have been a part of this experience. I first want to thank Michael Theodore for shining a light through his interdisciplinary work. You are the reason I was able to come to Colorado to do this work. I am grateful for all the conversations and encouragement to make these projects come to life.

It started at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford. The enthusiasm for exploration, high standards for technology design, research and dedication to music along with the warmth of the community surrounding CCRMA had a profound impact on me. Chris Chafe maintains a welcoming atmosphere for inspired computer music research collaboration. I'm honored to be a part of that community. Thanks to my creative and bright peers and friends I worked along side including Chris Carlson, Hunter McCurry, Mike Rotondo, Michael Wilson, and Eli Noune among others. Thanks also to the talented professors and instructors especially Ge Wang, Jonathan Abel, Dave Berners, Wendy Ju, Edgar Berdahl, Sasha Leitman, Jay Kadis, Jaroslaw Kapuscinski, Fernando Lopez-Lezcano, my advisor Julius Smith, and emeritus professors John Chowning and Max Mathews for their patience, dedication and inspiration.

My committee—Mark Gross, Tom Yeh, Mike Eisenberg, Miller Puckette, and Michael. An extra thanks to Mark for his thoughtful and frank feedback on everything from edits to big picture thinking.

Thanks to Alicia Gibb and all my great lab mates in the BTU Lab, Zack Jacobson-Weaver, Danny Rankin, Abhishek Narula, and others for creating a space where this interdisciplinary work fit in at the University of Colorado. I'm grateful for my friendships and collaborations with Hynjoo Oh and Abhishek Narula. Thanks also to

John Drumheller and John Gunther for always keeping the door open with BLOrk as a space for musical experimentation.

Thanks to Trimpin, for his earnest and open approach to art and teaching. I am grateful for the opportunity to have worked and learned along side him during a residency at both Stanford and the University of Colorado. I hope we meet again.

Thanks to the ATLAS institute and the National Science Foundation program eCSite for provided the funding that made this work possible.

Finally, I want to thank my incredible family: my mom for always instilling hope instead of fear, my dad, for encouraging me to solve problems and my sisters for making life fun. And to my growing family, my children, Penelope, Amelia and Otto, for bringing so much love and joy into my life, and to my wife Jamie for being my partner through it all.

Contents

Figures

Chapter 1 Introducing Scaffolded Design Tools for Interactive Music

1.1 Introduction

In music, and the arts, electronic and computational technologies have created new ways of working, enabling new forms of creative expression and new experiences. *New media artists* and experimental musicians are exploring the possibilities enabled by computational, interactive and robotic technologies, and digital fabrication tools to enable new forms of expression and to create new experiences for audiences. The artists and musicians who employ these technologies at the bleeding edge create new techniques, hardware and software. New mediums and approaches like augmented reality, 3D printing, interactive and kinetic sculpture, new interfaces for musical expression, and robotic musical instruments are nascent compared to the violin or oil paints. These mediums represent new opportunities that require new tools and approaches to realize their potential. This dissertation is about creating and using such *design tools for interactive music*.

In recent years creative applications of new technology have been amplified by the connectivity of online sharing platforms and social networks (for both finding inspiration and getting help) as well as new tools for creating and designing with new media, including open source software and hardware, and online tutorials. Many artists and musicians have found ways to incorporate new interactive technologies to enhance their practice. However, creating work with these emerging mediums requires a significant level of technical skill and new approaches that do not correspond with traditional arts training. Scaffolded tools that lower the barrier to working with these technologically rich mediums are needed to invite more participation, ideas and innovation. In order to have the greatest impact, these tools should be open and extensible, and they should support existing modes of working. By introducing such tools we create a pathway that allows a broad population to create in these mediums.

This dissertation explores approaches to broadening participation in technology based and interactive art and music through design tools I have developed for creating digital music instruments and for making and controlling musical robots. These tools are informed and inspired by research on the design of new interfaces for musical expression and the use of scaffolded design tools found in open-source hardware and software communities and educational settings. This dissertation describes the iterative design approach that led to my current designs including experiences in workshops using these tools with children and adults as well resulting instruments, performances

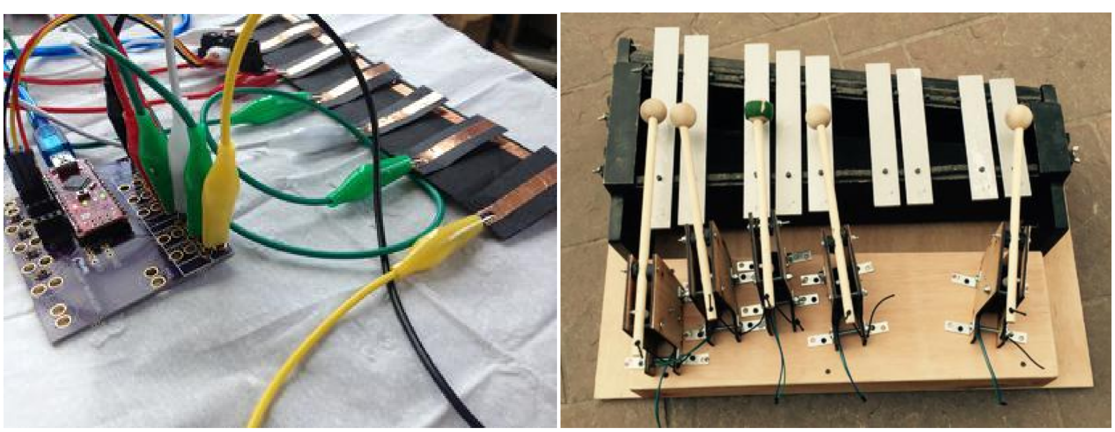

Figure 1.1 –Projects created with scaffolded design tools. A "paper piano" connected to the Sound Clippys Board (left); An Automated Xylophone created with the MIDI Bots platform (right)

and interactive sound art installations. I focus on two platforms: the Sound Clippys Platform that includes a hardware board and a software library for building Digital Music Instruments (DMIs), and MIDI Bots, a musical robotics platform that includes the MIDI controlled motor driver boards and a series of mechanical attachments for building digitally controlled sound actuators, or *musical robots* (Figure 1.1).

This chapter outlines the motivation and research goals of this dissertation. In Chapter 2 I present related work that informs my designs. This includes research on the design of new interfaces for musical expression with a focus on building DMIs and techniques and applications for sound actuation. I also review concepts and designs from research on scaffolded educational tools and propose a novel framework to apply to tools for designing digital musical instruments and musical robots. Chapter 3 details the foundational work I have done that has led to these platforms for interactive music. The remaining chapters detail my designs, their subsequent use and my findings.

1.2 New Media, New Possibilities

Computational and interactive technologies enable new possibilities for engaging with and creating music. Using computers composers can develop algorithms to compose and generate new music in real time. With interactive technologies performers can make instruments that can respond to physical gestures to control and shape sounds and trigger musical events. As a medium, interactive and computational technology offer seemingly endless possibilities. In order to leverage this potential, musicians need to shift from passive use and consumption of music technologies to the design and creation of music technologies. By empowering musicians with malleable tools that make the power of technology accessible for creative use we can reorient computational and interactive technologies from being tools to serving as a medium for creative expression.

Let’s look at how technology is commonly used in music today for the following categories: sound design, music composition, digital musical instrument interfaces, and digital sound production. While not representative of all musical practices these four categories represent a broad spectrum of uses of technology for music.

1.2.1 Sound Design and Composition

Digital tools for generating and controlling sound range widely. At one end of the spectrum easy-to-use tools tend to provide few parameters for customization in favor of an accessible out-of-the-box experience. These tools engage a broad population. They don’t require specialized training, and allow beginners to make coherent music. For example, entry-level production tools like GarageBand include loops of musical phrases, a collection of virtual instruments and audio effects to build tracks with pre-built ingredients. Meanwhile, in professional recording studios we find highly parameterizable instruments and racks of expensive external processing gear to customize and manipulate sounds. Different combinations of gear and parameters can produce unique sounds, but however well designed, these tools are static. They have functionality built into them that cannot be modified. If musicians instead learn the fundamentals of sound design through programming environments they can create sounds and audio effects that go beyond these fixed designs.

Tools are available for savvy musicians, who are ready to work at a lower level to customize their sounds, but there is a high barrier to entry. This changing landscape of technology for music has implications for music education. As tools for creating music become more entwined with digital means of production, musical training must embrace new skills. Fundamental knowledge of sound design, building and using audio effects, sound synthesis techniques, and working with and manipulating live audio and recorded audio samples gives musicians the ability to create and shape sounds. Knowledge of algorithmic approaches to composition can give composers access to an endless source of musical ideas. These algorithmic musical ideas are generated based on aesthetic choices made by the programmer/composer and high level parameters that influence the music being produced. Meanwhile, interactive technologies enable new ways of interacting with and controlling sound. The use of computation and interactive technology presents opportunities for today's musician that can only be leveraged with new approaches to education and tools for creating music with these technologies.

Audio programming languages have been created to help musicians and composers leverage computers for music. Pure Data (Pd) is one of those languages[1]. Pd is an open-source visual programming language for interactive audio and visuals. To help beginners get started programming for music I developed the Modular-Muse software library for Pd, the software component of the Sound Clippys platform. This library manages many of the complexities of getting started with sound design in Pd. It works inside Pd, so users are introduced to some of the ways Pd works with easy-to-use blocks that allow for a range of sound explorations. It also manages communications with an Arduino microcontroller to simplify the use external hardware to trigger and control sounds.

1.2.2 Musical Interfaces

With modern digital musical instruments and interfaces, approaches to controlling sound have remained stubbornly tied to traditional musical instrument paradigms. MIDI keyboards are the dominant interfaces used to control and create music with a computer (along with the standard QWERTY keyboard and mouse). The main advantage of using

[1] www.puredata.info

traditional musical instruments as models for digital musical instruments is that technique learned on a traditional instrument can be applied to its digital counterpart. However, unlike traditional music instruments, digital musical instruments are not bound to physical acoustic properties. Digital musical instruments can use a variety of increasingly available and inexpensive sensors to capture gestures from a performer that can map to any sound or musical parameter.

The small but growing field research of on *new interfaces for musical expression* (nime), explores a range of new approaches to building expressive digital musical instruments. There are many areas of future work within nime, including questions such as: can a digital instrument be easy to play, yet allow for the development of virtuosity? (Dobrian et al, 2006) What is the lifespan of a nime? (Berdahl & Ju, 2011) Can an instrument be created for a single performance and then taken apart or reconfigured? Can it be reconfigured as part of a performance? What does it mean for composers and musicians to build their own instruments? My Sound Clippys Board is designed to make it easy to begin down this path of digital lutherie, to explore these questions, and to inspire the next generation of artists and instrument builders.

1.2.3 Making Noise

In traditional instruments sounds are produced through the acoustics of the instrument. The performer bows a string and its vibrational energy is transferred to and amplified by a soundboard. By contrast, electronic and digital instruments use amplified loudspeakers to produce sound. Sound actuation is an alternative approach to producing sound acoustically through physical motion. This can be through elaborate mechanisms and with digitally controlled motors that tap, strike, shake or otherwise produce acoustic energy.

Sound actuation can augment existing instruments or enable automatic or robotic musical instruments. This enables the production and control of acoustic sounds under digital control. These techniques enable new sound experiences not possible with loudspeakers.

1.3 Digital Skills and Modern Musicians

Contemporary musical training should include skills that will enable musicians to design their own sounds, generate musical algorithms, and to develop digital music instruments and performances that allow them to interact with music in new ways. In a recent blogpost, nime researcher Alexander Refsum Jensenius contrasts the traditionally separate roles of *instrument constructor*, *musician* and *composer* with the increasing merging of these roles[2]. He describes the growing international community of researchers and artists where these roles merge and individuals, or small teams, imagine, design, create, compose for, and perform with a new instrument in quick succession.

Jensenius notes this merging is predominantly taking place outside of traditional music institutions. I view this as an opportunity to invite composers and musicians to engage in designing and integrating technology into diverse and creative musical practices.

Little has been written on the development of design tools for musicians that attempt to lower the barrier to designing for interactive music. The work described in this dissertation uses lessons from scaffolded educational and design tools for programming,

---

[2] http://www.arj.no/2012/08/16/roles/

robotics, and digital craft, to create new systems to enable a range of users to engage with music technology as designers.

Peppler & Kafai, (2005) argue that learning the language of creative coding (coding for creative expression) is essential to expression in any digital medium. They describe how treating coding as a form of artistic expression has implications for students' 21st century literacy in both art and technology. This dissertation expands on this idea, by including electronics and interactive technologies (such as sensors and motors), as relevant mediums for artistic expression.

The work in this dissertation is aimed at bridging the divide between traditional approaches to making music and those currently being explored in computer science departments, interaction design communities, interdisciplinary music and technology research centers, and high school robotics clubs. In this dissertation I show how my tools lower the barrier to emerging technologies, enabling beginners to participate in these practices. By creating different design platforms and using them in formal and informal educational settings, I have identified and addressed some specific challenges and identified directions for future work.

1.4 The New Media Shift

There are wide ranging reasons to make designing with technology more accessible for creative applications. Computational and interactive technologies are emerging mediums for so-called new media artists. However, designing with these technologies requires specialized skillsets traditionally taught in engineering and computer science programs. These skills don't overlap with traditional arts training. Tools that lower the barrier to entry enable a broader population to produce new media art.

A collection of articles in *Artnodes* titled "New Media, Art-Science, Contemporary Art: Towards a Hybrid Discourse", draw attention to the historical resistance to *new media art* by purveyors of contemporary art and the growth in activity and institutions with a focus on new media art (Shanken, 2011). The articles situate new media art in the larger context of modern contemporary art. For example, Allen (2011) argues that the recent attention paid to new media art through archival and curatorial efforts will serve in "securing [its] rightful detent as another lineal 'movement' in histories of creative practice".

> It was against what the department said music was; they said I was dehumanizing music! My response was, 'Perhaps it's the *humanization* of computers.'"
>
> John Chowning, 1964

Music has seen a similar resistance to the inclusion of computers. As indicated by the above quote, this was the case in 1964 with John Chowning at Stanford and continues today as indicated by Jensenius's post from 2012. As with new media art, interactive music has found other paths with thriving communities in electrical engineering, and computer science departments, and makerspaces. Perhaps most telling is the growing number of popular musicians and performers using novel technology tools as part of their artistic practices.

New media art and interactive music involvement is growing, and given the growing pervasiveness of technology, it is reasonable to assume activity around new media art will continue to increase. However, *new media art* participation is limited because of the need for specialized technical skills not traditionally found in music and arts training.

1.4.1 Computational and Interactive

Interactive technologies enable the creation of art that responds to our environment and our actions. Computation allows these interactions to possess intelligence, even if artificial, that changes our relationship with the art, allowing us to interact actively rather than passively. Traditional music instruments rely on the physical properties of tuned strings and soundboards, or the resonance of air pushed through pipes, etc. to produce sounds. Analog electronics and computation enable new means of producing sounds that no longer rely on these physical, acoustic properties. Such interfaces can mimic traditional instruments such as those on a electronic keyboard or digital drum set, or they can use new sensors such as touch screens or accelerometers.

Meanwhile robotic technologies and mechanisms enable new means of controlling and producing acoustic sounds that are difficult or impossible to create by human hands. Musical robots can be used to accompany other musicians, to augment an instrument or performers capabilities, or as standalone, controllable automated musical instruments.

These computational and interactive technologies enable powerful new paradigms for creating music and art. Imagine what we might see 20 years hence if students begin learning to leverage these technologies in their creative output, alongside traditional techniques taught in primary school.

1.4.1 Pushing Media Boundaries

An artist exploring and pushing the boundaries of a new medium often must learn new techniques or acquire new skills in order to explore new ideas and create within the medium. In technology based mediums, a visual artist might learn software programming or a sculptor learn about 3D printing. There are various tools available, and established online communities to leverage for various mediums. These include tools like Processing for programming visuals, Arduino for developing interactive works, and Pd for audio programming. Although targeted at artists, these design tools are created more in the vein of engineering design tools than consumer level digital design tools such as Adobe Illustrator or GarageBand, which may intimidate those without programming or electronics experience.

To make these activities more inviting *new tools are needed that bridge this gap*. By building transparent tools on top of or in conjunction with existing tools, the platforms and ideas presented in this dissertation offer a model for *scaffolded design tools for interactive music*. These tools lower the barrier to entry with scaffolding that supports the user in doing more than they could otherwise. These tools are open to a wide range of creative applications, and by working within existing ecosystem of tools they are extensible and provide the opportunity to transition beyond the particular confines of the tool.

1.5 Research Statement

Combining knowledge of interactive music, interaction design can enable the development of tools and activities that foster creative applications of technology, creative expression and informal learning.

This work introduces new approaches that lower the barrier to working with interactive technologies and programming in the context of music, specifically for designing digital musical instruments, and for building automated musical instruments.

The goals of this research are as follows:

(1) Develop easily apprehensible yet flexible and malleable tools to support a range of interactive music design activities inspired by relevant works and research on design for interactive music. The design of these tools is informed by interactive music research and scaffolded educational tools.

(2) To use and study the resulting tools, including the decisions and iterations that led to the current designs, and how they are used in workshops, performances, and art installations. The development and subsequent workshops and applications of these tools serve as case studies in the design of *scaffolded design tools for creative technology*.

(3) Identify the affordances of these tools through the projects built with them as well as potential future work with these platforms. Identify the weaknesses and limitations and possible alternatives to my approach.

(4) Identify a common framework to propose a model for creating scaffolded design tools for interactive music.

1.6 Approach

The work presented in this dissertation uses formative evaluation in the development of hardware, software and physical designs that have been refined over the last few years. The tools described here have evolved through a series of iterations in response to experiences teaching with them and using them. The development of my own instruments and art works has given me personal experience with the fields of digital musical instrument design and the use of sound actuation.

Another way to frame my approach is with *research through design,* a framework that uses the lenses of *process, invention, relevance*, and *extensibility*. According to Zimmerman et al, "following a research through design approach, designers produce novel integrations of HCI research in an attempt to make the right thing: a product that transforms the world from its current state to a preferred state." (2007) Zimmerman and colleagues identify four "lenses" for evaluating a research contribution:

Process - In documenting the contributions, the researcher should include enough detail to enable others to reproduce their designs, and provide rationale for their specific design decisions (Chapters 3, 4, and 6).

Invention - The researcher must demonstrate that they have produced novel integration to address a specific situation (Chapters 4, 5, 6, and 7).

Relevance - Researchers must articulate the prefered state their designs attempt to achieve and why this state is preferred (Chapters 2, 3).

Extensibility - The design research has been designed in a way that the community can build on and leverage the knowledge derived from the work (either the specific design, or ideas that can be applied in a different space) (Chapters 4, 5, 6, 7, and 8).

1.6.1 Design Genesis

My initial inspiration for this work came at the 2013 TEI doctoral consortium where I presented at the doctoral consortium. The mentors at the consortium were Tom Moher, Yvonne Rogers, Mike Horn, and Leah Buechley. Through conversations and ideas that coalesced at the conference I was inspired to develop a toolkit for interactive music. I wanted to make it easier for a broad audience to be able to develop new interfaces for musical expression, using sound design, interface design and sound actuation. Since then I have developed and iterated this idea. This has led to two platforms: Sound Clippys and MIDI Bots.

The work presented in this dissertation is exploratory. This work draws on research on interactive music, interaction design and scaffolded educational tools. I draw on ideas from projects and research across interactive music and sound to define my design goals. This dissertation frames the resulting activities and artifacts within the context of interactive music. My approach to tool design and teaching draws on projects and approaches from constructionist environments where people *learn by doing* and where *creative experimentation* is an important process. Specifically I draw inspiration from *making* and *digital craft* communities and the creative combination of digital and physical found there.

I will show how these areas have influenced my hardware, software, and physical designs. In particular I show how my solutions to specific challenges of *making* with hardware and software demonstrate new and productive forms of support.

My designs have evolved through cycles of deployment and redesign. I detail my designs and how design decisions relate to my criteria. The description of how these tools have been used and the resulting interactive music projects that have been created using my designs are used to identify strengths and weaknesses.

1.7 Contributions

The key contributions of this work include my hardware and software platforms for supporting digital musical instrument design and creating and using musical robots. I have explored these platforms through workshops, new instruments, musical performances, interactive sound art installations.

This research illuminates some of the challenges, key features and potential of *scaffolded design tools for interactive music*. The tools and techniques described here represent alternative means of creating inputs and outputs. This work aims to expand awareness of these alternatives and lower the barrier to using these approaches. Additionally, I hope to inspire others to build *scaffolded design tools for interactive music,* and more generally for *creative technology*. My hope is that this work helps to increase our capacity to invite and support a broadening audience of designers for interactive music.

The two platforms: Sound Clippys, and MIDI Bots, contribute specific techniques for supporting activities for interactive music. These platforms introduce new approaches to lowering the barrier to working with interactive technologies, allowing them to become relevant mediums for musical expression.

The final contribution is the development and application of my design criteria for *scaffolded design tools for interactive music*.

1.7.1 Scaffolded Design Tools for Interactive Music

I have identified three key design criteria for creating scaffolded design tools for interactive music. These criteria were used in the development of my design platforms. I show how these criteria are manifested in my platforms and how they influence the subsequent use of my platforms in a variety of settings that include educational and the development of new creative works. My design criteria of *support*, *compatibility*, and *versatility* are detailed in Chapter 2.

The results of workshops, performances and art installations show how my platforms enable multiple points of entry, allowing beginners to engage with different aspects of interactive music design, and support experienced users in doing more. They demonstrate compatibility with existing tools and ways of working, allowing a pathway to expertise. These design criteria promote iteration and creative exploration of techniques and concepts used by new media artists and musicians today.

This work aims to expand the vocabulary of music, not just by introducing new techniques and tools, but by inviting more people into the conversation. Tools for interaction design and controlling motors are not new—this work focuses on expanding the potential for interactive music by scaffolding the use of these technologies for a diverse set of users. The tools presented in this dissertation lower the barrier to entry for creating music and art with interactive technologies.

Chapter 2 Related Work

A number of fields inform and inspire my ideas and designs, including research on the design of new interfaces for musical expression, the use of sound actuation in interactive sound art, automated instruments and musical robotics. In addition to these areas of computer music, research and design of tools for education and prototyping and approaches found in constructionist learning environments influence my work.

The premise of this literature review is that lowering the barrier to designing for interactive music has *implications for musical creativity.* These activities also hold great potential as a *context for hands-on STEM education* through personally motivated design projects. Activities like the ones described in this dissertation that combine music and technology design, engage makers in both analytical and creative thinking. Building digital music instruments engages makers with circuits, sensors, microcontrollers and programming along with musical concepts such as scales, timbre, and sound design. Working with musical robotics exposes makers to mechanisms and motors along with acoustics and music programming. The *scaffolded design tools for interactive music* presented in this dissertation lower the barrier to creating and using digital music instruments and musical robots.

I first overview work in *Computer Music and Interaction*, and *Automated and Robotic Instruments*. Next, I highlight approaches to teaching through design, creativity support tools, and educational design tools with a focus on constructionist activities, characterized by “hand-on explorations that fuel the constructive process” (Ackermann, 2001). I draw on insights from this research and consider how they might apply to building tools for making DMIs and musical robots in order to lower the barrier to working in these domains. I summarize these results with my design criteria for *scaffolded design tools for interactive music*.

2.1 Interactive Computer Music

In *The Digital Computer as a Musical Instrument*, Max Mathews proclaimed that computers could be used to produce any perceivable sound (Mathews, 1963). He describes a system of “unit generators” each with its own function can be combined in different ways to create a range of sounds. Matthews had already developed the first programming language for synthesizing sound, simply named MUSIC. In this article he also introduces how a composer can control the parameters of a sound synthesis model by issuing instructions, as with a traditional music score, to make changes throughout a composition. Further he discusses using the computer as a compositional aid, in which algorithms generate new musical material, which a composer can work with and develop further. Today Mathews’ *unit generator* approach to sound synthesis and processing audio is still used today in audio programming languages Max/MSP, Pd, ChucK, Supercollider and others. A unit generator can generate an audio signal or modify one. Control parameters allow the programs to shape sound programmatically or by allowing a performer to interact directly to enable new musical interaction.

Digital sound synthesis has become increasingly accessible on smaller and faster computers. Today, even inexpensive computers (such as the Raspberry Pi or BeagleBone Black) enable real time control and interaction with sound. In addition to introducing computer based approaches to sound synthesis and music composition, Max Mathews also explored new interfaces for musical expression. He created an interface called the Radio Baton that enables a conductor like interaction where two batons are used to play a musical score (Figure 2.1). Today, Mathews' vision of leveraging computation for composition and performance is possible with audio programming languages and is being pursued and pushed by a relatively small, but growing group of musicians and composers.

However, despite the availability of computers, inexpensive sensors and microcontrollers, and programming languages for music, most music students remain unfamiliar, and unable to leverage the untapped possibilities. Instead, there has been a

Figure 2.1 - Max Mathews with his digital musical instrument the *Radio Baton*

convergence around the use of computers in the production of music that has narrowed the scope of what a computer means for music making among most musicians and the general public.

To explore beyond currently entrenched approaches for using computers in music to the possibilities Mathews imagined, requires a pathway to these alternatives that is accessible to a wide audience. Using computation for music enables new ways of making music, and thinking about musical ideas like rhythm, harmony and melodies. Music theory, and approaches to music composition both use patterns, repetition, rules, tendencies, and variations. These are concepts that are well suited to computation.

2.1.1 New Interfaces for Musical Expression

Research on new interfaces for musical expression (nime) focuses on new means to create and control sound and music. I give an overview of this field, including challenges, opportunities and current research directions. Central concerns include playability, virtuosity, expressivity, and the opportunities in alternative interfaces.

Physical Gestures and Musical Expression

Traditional music instruments are inherently physically embodied. It takes physical effort to play them, as a musician moves his or her body to manipulate the instrument and thereby control pitch and timbre. The amount of energy exerted typically has a direct effect on the volume and brightness of sound produced. This physical relationship is apparent—it can be seen and easily understood. An expert musician plays with accuracy, expressivity and nuance, demonstrating control over the connection between gesture and sound. Attempting to leverage these affordances, musical interface designers have invented a wide range of controllers for digital music instruments. Some of these have acoustic counterparts while others introduce wholly new ways of creating and controlling sound.

Alternative Interfaces

Most people are familiar with instrument-like controllers that emulate the piano keyboard or wind instrument interfaces. These instrument controllers allow traditional technique to be transferred to a digital instrument. The exploratory side of this field has pioneered new interaction paradigms enabled by technological developments. For example, enabled by advances in radio technology in early 20^{th} century, the Theremin allows gestures in mid-air to control pitch and volume (Theremin, 1928). In 1984, Michel Waisvisz's experimental music controller "The Hands" was one of the first to utilize the MIDI standard, allowing his hand-held interface to control any MIDI controllable sound parameters (Figure 2.2). In Waisvisz' words it allowed him, "to walk, move and even dance while making electronic music. This was such a liberating prospect after having had to work with big chunks of unmovable analog electronica tied up in the early

Figure 2.2 - Michael Waisvisz playing *The Hands*

electronic music studios."[3] These early examples demonstrated novel interactions to control sound not found in traditional instruments. Because electronic and digital sound is not reliant on physical acoustics (beyond the function of a loudspeaker), control interfaces for these instruments are free to map arbitrary gestures to sound.
A project called Touch Keys augments a digital piano keyboard with capacitive touch sensitivity to detect relative position and movement of the fingers on the keys (McPherson, 2012). This enables new techniques for keyboard players. A vibrating finger controls vibrato similar to string instruments while vertical motion allows pitch shifts for each individual note. The Touch Keys technology opens up new musical possibilities by allowing a keyboard to respond to more than the force on a struck key.

The Wekinator, created by Rebecca Fiebrink, is a software layer that supports the development of new musical interfaces (Fiebrink, 2009). It supports users in using machine learning to train a musical instrument. The Wekinator translates input gestures into control messages that can be mapped to a software instrument. A user trains the system by repeatedly performing a gesture that can then be assigned to control specific sound parameters. Her tool is designed to lower the barrier to using machine learning for musical control. This open ended tool for music is an example of what I call *scaffolded design tools for interactive music*. The Wekinator lowers the barrier to working with machine learning in the development of digital musical instruments. It can be used for a variety of applications and it is compatible with existing tools including input devices and audio programming environments.

These examples represent a range of approaches found in the design of new DMIs. Perhaps most telling is the fact that these technologies and new approaches to making music are increasingly being used by professional musicians. These new interfaces for

[3] http://www.crackle.org/TheHands.htm

musical expression enable performers to create compelling performances that use physical gestures an audience can see, as opposed to micro-movements happening

Figure 2.3 – Digital musical interface designs used in music performance: Imogen Heap with her interactive gloves (left); The Reactable, used on tour with artist Bjork (right)

behind a laptop. Two such performers are Bjork and Imogen Heap. Bjork toured with a tangible tabletop interface for music called the Reactable (Jorda et al, 2007). The Reactable uses physical tokens that can be moved around on a tabletop surface. Computer vision identifies fiducial markers of the objects on the table to track their location and orientation. Another artist, Imogen Heap, formed a team to develop a new set of gloves she could use for performance[4]. (Figure 2.3) This collaboration resulted in *The Gloves*, which she uses as both an instrument and a controller for a popular music production and performance software called Ableton Live[5].

2.1.2 Sound Design

Sound design is the process of specifying and manipulating sound parameters for sound synthesis. Common digital sound synthesis techniques are available in a wide range of tools and approaches. Often sound design is done with high level commercial tools. These so-called software synthesizers provide high level access to sound parameters.

[4] http://www.imogenheap.co.uk/thegloves/

[5] https://www.ableton.com/

Sound design is also commonly done in music centric programming languages such as ChucK[6], Pd, or Max/MSP[7]. These languages scaffold many of the common approaches to sound design with building blocks that can be combined in different ways, providing more flexibility than software synthesizers. To have full control over bit and sample level details of a sound design algorithm requires a low level programming language such as C++.

2.1.3 DMI Design and Evaluation

Malloch et al identify a defining feature of DMIs as the separation of the human interface and sound production (Malloch, 2011). In designing a DMI the *primary task is mapping this relationship.* For traditional instruments, the mapping of inputs to sound parameters is physically bound. For instance, the fundamental pitch of a given string can only be altered by changing the tension or by changing the length. However with digital music instruments there is no such required mapping and instead we have the freedom to design the relationship between physical action and sound production. This allows consideration of other factors like ergonomics (comfort and convenience), performance (what will be interesting for an audience), playability (a digital instrument can be "retuned" to only allow the correct notes for the current setting), etc.

Cook (2001 & 2009) offers guiding principles to consider in the design of new instruments. Several of his principles are highly relevant for introducing NIMEs to beginners. These include: Music+Science is a great teaching/marketing tool and the younger the student the more fearless. 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 novice builders. The suggestion that

[6] http://chuck.stanford.edu/
[7] https://cycling74.com/products/max/

everyday objects suggest amusing controllers gets at the playful potential of designing DMIs and is likely to be attractive to children.

Jordà and Mealla (2014) introduced a method for teaching DMI design focusing on the challenges of *expressiveness* and *mapping*. *Expressiveness* is about creating an interface that allows a performer to express or communicate emotion through music. *Mapping* is about designing the relationship between the instruments inputs and sound. They found participants who had previous musical knowledge when evaluating their DMIs had a better grasp of the concepts of expressiveness and virtuosity. This suggests a potential advantage in teaching nuanced concepts of interaction design for music to students with musical training, another motivation for engaging musicians in designing DMIs.

Verplank, Sapp, and Mathews describe the theme of *buttons and handles*, used in one of the first NIME based courses taught at CCRMA to teach controller development (2001). Buttons are for triggering events or changing state, while handles allow continuous control. *Buttons* and *Handles* are part of Verplank's framework for interaction

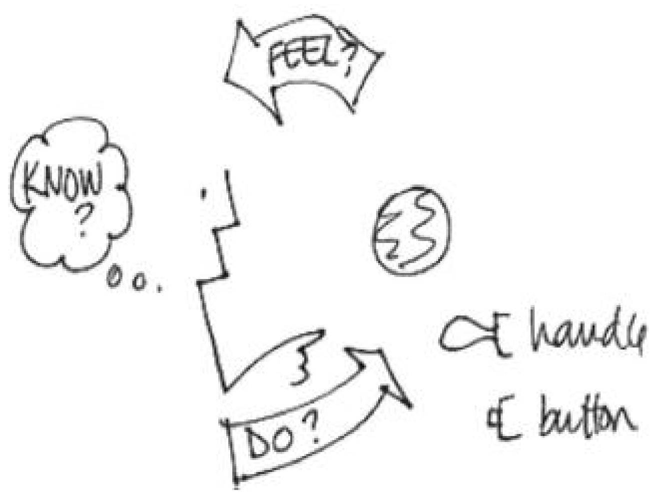

Figure 2.4 – Bill Verplank's questions for interaction designers

design[8] (Figure 2.4). Through hands on activities students develop practical skills, culminating in the development of new interfaces with a wide variety of novel designs with potential applications well beyond music, concluding that “the direct engagement in an expressive realm like music can generalize to a wide range of human-machine controllers”. The implication is that skills and techniques learned in designing DMIs are transferrable to other domains.

Verplank’s notion of *buttons* and *handles* provides a framework for thinking about discrete (digital) vs. continuous (analog) control, both of which are critical in the design of expressive digital music instruments. The concepts of *expressiveness* and *mapping*, and *buttons* and *handles* inspired the design of the Sound Clippys Board and the Modular-Muse Library. The Modular-Muse Library provides sound design tools for building expressive sounds with different parameters and audio effects that can easily be mapped to digital (buttons) and analog (handles) inputs on the Sound Cippys Board. This platform allows the development of novel and personally meaningful interfaces with understandable but varied mappings. It provides a pathway to the sophisticated ideas put forth by Cook by modest steps into this field. In chapter 4 I detail the design of this platform and how I and others have used it to create DMIs.

2.1.3 Sound Actuation: Automated and Robotic Musical Instruments

This section presents historical examples of sound actuation, from mechanical systems to digitally controlled musical robots. Recent developments and applications of sound actuation inspire the development of the MIDI Bots platform. We will see how digital control of physically actuated sounds enables the expression of new musical ideas.

[8] https://ccrma.stanford.edu/courses/250a-fall-2004/IDSketchbok.pdf

Sound actuation and the development of musical robots holds great potential for developing innovative musical instruments and experiences. Traditional musical instruments produce sound through acoustic energy from vibrations in the instrument itself, whether a bowed violin string, a struck drum head, or a resonating column of air inside a clarinet. The sound comes from the location of the instrument, not from a physically separated loudspeaker.

Harmonically rich sounds and variations in timbre from acoustic instruments arise from complex interactions between the performer and the materiality of the instrument. The pursuit of such richness in sound texture and localization in space has occupied the attention of much of the computer music community for decades. This has yielded sound synthesis and sound spatialization techniques using analog and digital approaches and loudspeaker arrays. The physical properties of instruments can be modeled and sound sources placed virtually in space, but these are always approximations. By contrast, sound actuation uses the physical properties of objects to produce acoustic sound, as with traditional instruments. In fact, often they are traditional instruments, augmented with actuators to play them automatically. In this way, automated instruments can leverage centuries of innovation in the design of traditional music instruments.

A Brief History of Sound Actuation

There is a long history of musical robots and their purely mechanical predecessors: music boxes, player pianos, orchestrions and the like. The modern history of sound actuation for music can be traced to mechanically automated instruments, which use pneumatic devices for actuation, including player pianos and the more elaborate orchestrions of the 19th and 20th centuries. These mechanical instruments became popular in the 19th century as a means of music reproduction; using cylinders, discs and paper rolls as memory (McElhone, 2004). As loudspeaker and recording technologies

Figure 2.5 - Automatic instruments by the Logos Foundation

advanced, these mechanical devices fell out of style (Murphy et al, 2012).

In the 70's and 80's new electronic means of automation to control sound actuation were explored by early innovators of musical robotics, Trimpin and Godfried-Willem Raes (Murphy et al, 2012)(Focke, 2011). Trimpin, inspired by the mechanical player piano, developed robotic membranophones and idiophones, as well as an 88 key piano robot, originally developed to play the music of digitized Nancarrow compositions rescued from decaying player piano rolls (Leitman, 2011). Raes' early works, by his own admission, were "soundsculptures in the full sense: not real musical instruments, and not playable"[9]. Today, Raes' Logos Foundation claims to be home to the world's largest robot orchestra[10] (Figure 2.5). They use their orchestra as a medium for composers and artists to work with to develop new concert experiences. In 2015 Raes built instruments

[9] http://www.logosfoundation.org/g_texts/ibart-leonardo.html
[10] http://logosfoundation.org/mnm/index.html

for electronic musician and composer Aphex Twin, who subsequently released an album called “Computer Controlled Acoustic Instruments Pt2” using the instruments[11,12] (Figure 2.6)

Meanwhile, Trimpin continues to develop and perform using sound actuation with his installations and performances, appearing internationally in museums and concert halls.

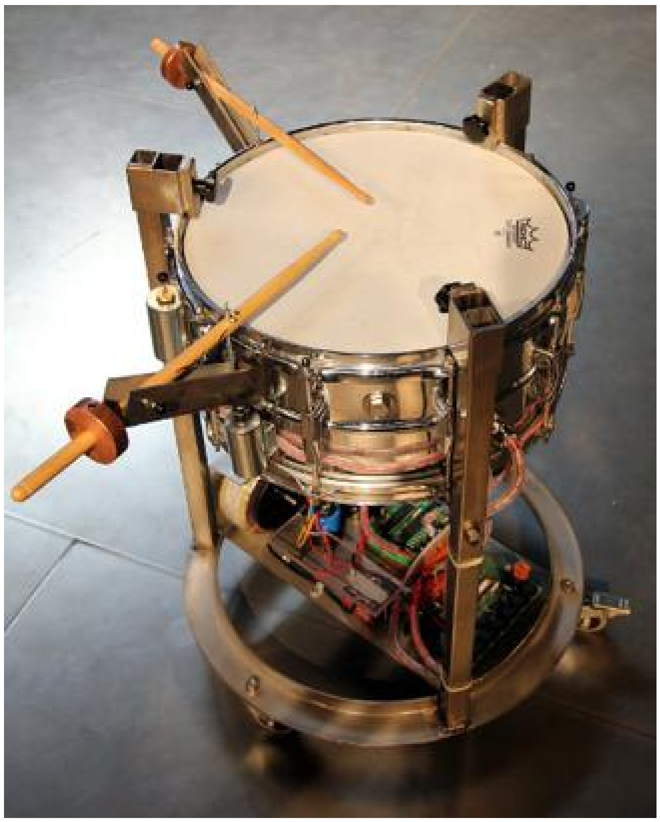

Figure 2.6 - Raes' <Snar_2>, built for Aphex Twin has two external drum sticks and many more internal to the drum

ATLAS’s own Center for Media Arts and Performance (CMAP) hosted Trimpin in November 2015 for a week of mentoring and constructing instruments, culminating in a performance at the Black Box Theater. The November 2015 performance featured many of Trimpin’s inventions along with my own designs.

[11] http://logosfoundation.org/instrum_gwr/HAT.html
[12] http://logosfoundation.org/instrum_gwr/snar2.html

Modern Musical Robots

Recent developments in musical robotics exemplify recent advancements and a range of musical styles where they can be found along with a variety of control and interaction paradigms. Beginning in 2000, Eric Singer's LEMUR instruments demonstrated how robotic instruments can be used to "augment the possibilities available to players and composers" (Singer, 2004). In 2010 jazz guitarist Pat Metheney collaborated with LEMUR to develop a large robotic band known as the Orchestrion to back him on tour[13] (Figure 2.7). The results of this collaboration demonstrate the musical potential of a large scale robotic orchestra developed for professional musicians. The artist Jazari uses an Arduino controlled series of instruments he created with impressive range. He interacts with his ensemble of automatic percussion instruments using techniques that borrow from electronic music. Others have explored intelligent interaction systems between musician and robotic performers, (Weinberg, 2006; Kapur, 2009). Recent research

Figure 2.7 – Guitarist Pat Metheny, and the *Orchestrion*

[13] https://www.soundonsound.com/sos/apr10/articles/orchestrion.htm

directions in this field include exploration of actuation strategies (Kapur, 2007; Rogers et al, 2015), and quantitative methods for musical performance evaluation (Long et al, 2015).

The ongoing activity and research in this space is promising. However, I believe that much is to be gained by making these technologies and approaches to music making more approachable. My MIDI Bots platform uses standard MIDI messages allowing control with off-the-shelf music software. Lowering the barrier to working with motors for musical purposes invites more musicians and roboticists to design musical robots to explore new applications of this technology.

Actuated Instruments

Another exciting direction in this field is in what Sheffield calls “performable actuated instruments” (2015). Such instruments are augmented with sound actuation to allow a musician to play and interact with acoustic features of an instrument alongside sound actuation. These performable actuated instrument enable complex musical interactions and novel expressive possibilities. Berdahl’s Haptic Drum showed how with a motor and a sensor a drum can “hit back,” enabling techniques not possible with a passive instrument such as a one handed drum roll (Verplank & Georg, 2011). Dan Overholt showed how the sound source for a digital instrument can be co-located on the instrument itself with the use of tactile transducers mounted to the body of the instrument (2011).

Approaches using electromagnets show how actuation provides an alternative to plucking, striking or bowing to induce vibrations in strings. The Electromagnetically Prepared Piano, Magnetic Resonator Piano, and the Magnetic Rhodes Piano introduced new systems and control strategies for actuated instruments (Bloland, 2007; McPherson, 2010; Shear & Wright, 2011). These instruments excite the strings, or metal tines in the case of the Rhodes, using electromagnets, allowing them to resonate without being physically struck. The use of actuation in these instruments expands the range of sounds and means of control beyond what is possible with the traditional approach that relies on the mechanism behind the keys. My own Feedback Lap Steel Guitar (Harriman, 2015) used a tactile transducer mounted at the bridge to actuate the strings of the instrument by physically vibrating the bridge. The Feedback Lap Steel demonstrates an inexpensive approach using off the shelf tools to incorporate sound actuation into a new instrument. Some commercial products have been introduced using electromagnetic sustain on an electric guitar such as the EBow, Sustainiac, and Moog Guitar.

Teaching Musical Robots

Kapur et al present *The KarmetiK NotomotoN: A New Breed of Musical Robot for Teaching and Performance*, as a musical robot for plug and play use (Kapure & Murphy, 2011). Using a MIDI interface to control up to 18 motors attached to a drum enables a

Figure 2.8 - Ajay Kapur's Karmetic NotomotoN

range of expressive sounds (Figure 2.8). They describe how the system allows material and mechanism explorations. The use of MIDI in this system makes it accessible to musicians and composers without a technical background. They use it as a platform for learning how to compose for musical robots and for trying out different striker mechanisms in a compact instrument.

As with The KarmetiK NotomotoN, I aim to create a low barrier to entry tool which allows quick and varied explorations of sound actuation. Some key differences in my approach are the use and exploration of found objects for sound, varied use of motors, and accessible techniques for building sound actuation mechanisms. I created MMMD to enable wide explorations with different types of sound actuation, to simplify design and control for composition, and to allow for a pathway to expertise.

2.2 Educational Tools and Learning Through Design

Designing for interactive music, specifically designing and building DMIs and musical robots, is an interdisciplinary endeavor. It involves programming, electronics, interaction design and a heavy dose of music. In order to develop criteria for tools that support this activity, I take a broad look at research on enabling design, supporting creativity and supporting novices.

I present three key design features (*supportive*, *versatile* and *compatible*) that I used in creating tools for designing DMIs and musical robots. Chapters 5 and 6 show how these principles are embodied in the designs and how they succeed or fail in use.

2.2.1 Creativity Support Tools and Construction Kits for Kids

Popular platforms and toolkits for children and those created for artists share similarities in approach. Resnick & Silverman (2005) outline their design principles for designing construction kits for kids. There is much overlap with a list of design principles for creativity support tools including: *low floor, high ceiling and wide walls*, *support many paths and styles, support open interchange, make it as simple as possible*, and *choose black boxes carefully*. According to Shneiderman (2009) *creativity support tools* should enable discoveries and inventions throughout a design process. He argues they should

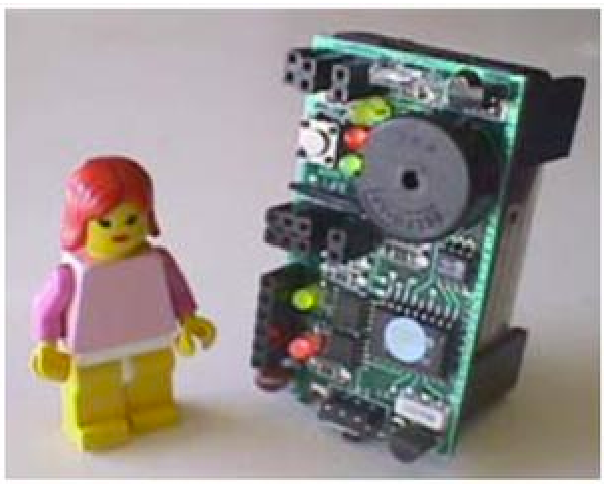

Figure 2.9 - The Cricket, from the Beyond Black Boxes Project

support rapid exploration and easy experimentation. Resnick and Silverman advocate designs where *a little bit of programming goes a long way*. They describes a shift in focus from depth of computer science skills to constructs kids seem to be able to easily grasp, allowing users to do a lot with a little. These approaches support users in doing more than they could otherwise, while promoting versatile applications.

An example of this work on building construction kits for kids is the *Beyond Black Boxes (BBB)* project where Resnick, Berg & Eisenberg (2000) describe how the electronic devices known as *Crickets* can be used to engage children in science in new ways, through *making* scientific instruments, not just using them (Figure 2.9). These kits provide scaffolding around electronics and programming to allow the users to do more than they could otherwise. This experience of designing scientific instruments has a relation to practice, putting kids in a designer role. Similarly, my work encourages musicians and artists to understand how the technology works in order to design and make DMIs and musical robots, rather than use off-the-shelf instruments.

2.2.2 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. (Reiser and Tabak, 2015) Scaffolding, which could be a mentor or a tool that guides the student, is common in children's programming languages such as Scratch (Resnick, 2009) and Etoys (Kay, 2005), which prevent illegal coding constructs. Hmelo and Guzdial compare *glass box* and *black box* scaffolding (1996). Glass boxes provide visibility into the inner workings of the scaffolding. Meanwhile, black boxes can be used to support a user without revealing how the scaffolding works. Hmelo and Guzdial argue, if you want the user to be able to eventually remove the scaffolding (often called fading), than a glass box may be more appropriate.

My designs use both black box and glass box scaffolding to support specific tasks in building DMIs and musical robots.

2.2.3 Making, Tinkering and STEM

Design tools for education will increasingly play a role in education as hand-on learning continues to see increased attention. Studies of constructionist learning show the efficacy of environments such as “computer clubhouses” (Kafai et al, 2009), a precursor to today's public maker spaces where young people engage with programming, electronics, and digital fabrication. These activities (Honey and Kanter, 2013) 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 recent K-12 consensus document *A Framework for K-12 Science Education (Quinn & Bell, 2013)*. Vossoughi and Bevan (2014) highlight ways that *making* and *tinkering* activities support learning by contextualizing STEM concepts in meaningful activities and cultivating interdisciplinary practices.

Petrich et al (2013) at the San Francisco Exploratorium’s Tinkering Studio identify the acts of getting stuck, iterating on a problem and solving it as a salient process of tinkering. Resnick and Rosenbaum (2013) reiterate 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.” The value in these spaces is in the way kids are working and how it relates to “professional practice”, getting stuck, solving problems and collaborating on personally meaningful projects.

2.2.4 Bricolage

Turkle and Papert contrast the programming styles of *planners* and *bricoleurs* (1991). The planner's approach is structured, breaking the problem down into submodules and systematically solving them. Whereas the bricoleur may start out with a goal in mind, the way it is achieved is through “collaboration with the machine”. Trying things and responding to the results.

McLean details how bricolage programming is particularly apt for artistic endeavors involving tools common to building DMIs such as Pd, Max, ChucK and SuperCollider[14] (2010). 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 (2009). This synergy of working styles creates a unique opportunity to promote the design of DMIs as a viable approach to engaging children in technology design (Figure 2.10).

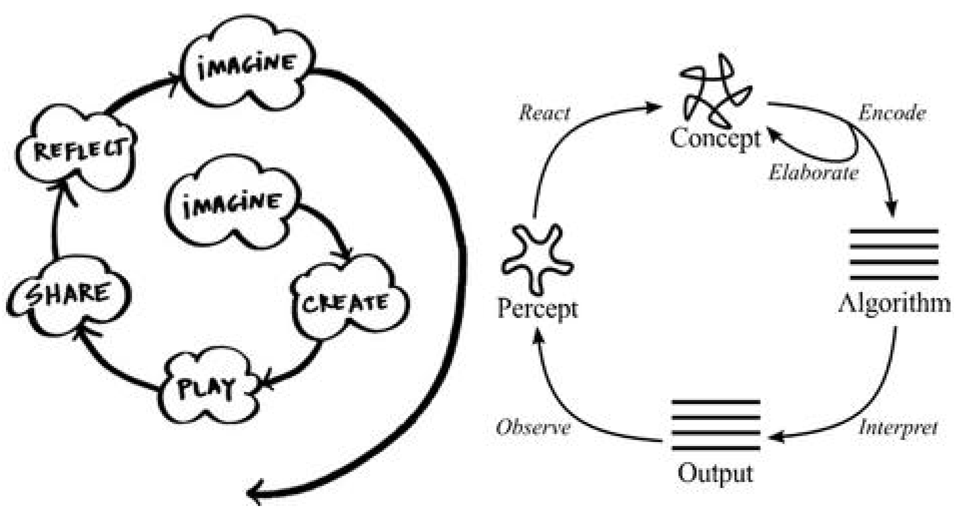

Figure 2.10 - Models for creative design: Resnick's Creative Thinking Spiral (left); Process of action and reaction in bricolage (right)

[14] http://supercollider.github.io/

A wide range of contexts and approaches are used to engage students in “learning through design”, with a variety of design tools. Next, I overview some of the most common approaches and identify lessons that can be applied to design for interactive music.

2.2.5 Engaging Programming

A child’s first encounter with computers is often through play. Inviting children to become designers of educational games, Kafai (1995) shows how they build deeper connections with underlying learning goals. Similarly, Webb, Repenning & Koh (2012) describe a guided discovery based pedagogical approach, teaching computer science principles through game design. This approach of leveraging personalization to increase ownership and motivation is a good fit with designing for interactive music where students choose the interaction paradigm, physical layout and materials, sound, and compositional components.

2.2.6 Hands-On Hardware

In holding workshops to introduce children to electronics through robotics, Rusk et al (2007) found that different students are attracted to different types of robotic activities, hence the importance of multiple pathways to engagement.

Blikstein and Sipitakiat (2011) argue for critically considering the affordances of various microcontroller designs for children to enable an appropriate level of transparency (support) for the audience and desired learning outcomes. In teaching robotics to middle/high schoolers and undergraduates, they identify challenges encountered by students using a “breakout” approach. These include: 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. They describe an alternative, which they refer to as the “cricket” approach. Named after the Cricket (Figure 2.9), this approach uses a series of modules, which abstract the complexities of building a circuit. Modules using this approach are easily connected with a single cable.

2.2.7 Music and STEM

Some have explicitly 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 (2012) identifies “musical playfulness” as a key to success, arguing that play is an important way children learn about the world. Trappe concludes, "music controller construction is a rich field to be discovered for pedagogical purposes” and a context that nurtures self-motivated creation, exploration and play. This dissertation builds on this idea, showing how the broader field of interactive music provides a productive context for design based learning and that scaffolded design tools can support this participation.

A study by Bukvic et al (2012) describes the use of *granularity* as a means to provide multiple points of entry, and in enabling an adaptive tool (Pd-L2Ork) that can match the educational model and skill level of the audience. I view this concept as a powerful approach to supporting multiple points of entry and the eventual fading of scaffolding.

2.2.8 Synthesis of Research on Educational Tools and Contexts

Through research on building tools for designers and supporting constructionist activities, I have identified a series of interrelated design criteria for evaluating and framing my own designs. These criteria are: s*upportive, compatible*, and *versatile.*

Supportive design has to do with scaffolding, which can be transparent (glass box) or opaque (black box). A versatile scaffolding can support a user to do a lot with a little, enabling them to do more than they could otherwise, but without limiting what they do.

Compatible design enables the transfer of knowledge and skills both in and out of a new tool. An adaptive tool enables working with the existing ecosystem of tools and techniques. They are adaptive and extensible to a variety of approaches and ways of working. It is important to note that compatibility is relative. That is, a tool may be compatible with one field but not another.

Versatile design promotes personalization. By creating flexible tools that support diverse outcomes, the user is able to express their own ideas. Versatility also relates to the target audience, is it for beginners or experts? Artists or technologists?

With these criteria in mind, I now look at existing tools that can be used for interactive music.

2.3 Tools For Interactive Music

Digital music instruments have an interface, a sound design and a mapping between the two. Meanwhile, musical robots have an actuator, some way of applying that actuator to a physical object, and a means of control. I will overview common approaches used today in designing for interactive music. From my research and experience there are more examples of tools to support creating digital music instruments than musical robots.

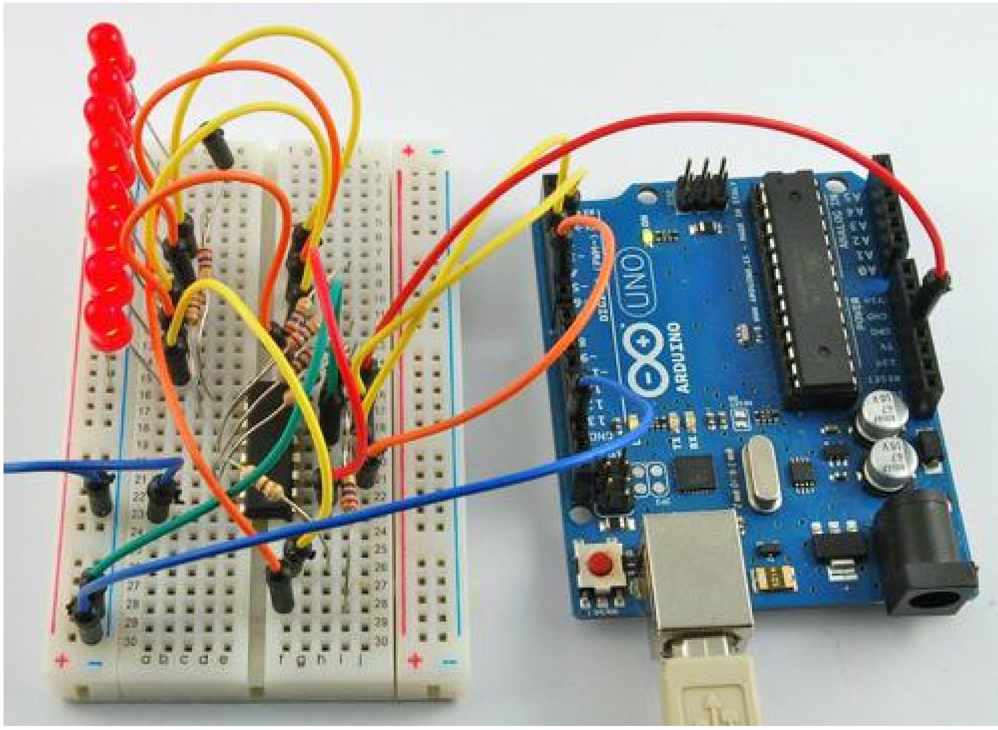

Figure 2.11 - The "breakout model" of interaction design, an Arduino and breadboard circuit

In describing a class at CCRMA, Wilson et al (2003) answer their own question of “Why Microcontrollers?” with *pedagogy*, arguing it provides the opportunity for students to learn about things like programming, digital logic, and A/D conversion among others. A typical approach to designing and building for interactive music uses an Arduino, and a breadboard with jumper wires and components that can be built into circuits (figure 2.11). This “breakout” approach is also common to teaching physical computing (Blikstein & Sikitipiat, 2011) and is highly versatile and compatible with a range of electronics. However, the breakout approach is not supportive for novices, and it is not compatible to typical ways of working with music.

Another approach is the use of toolkits with predefined modules and keyed connectors that only allow legal connections. These toolkits are closely related to what Blikstein and Sikitipiat call the “cricket” design. Because of the use of application specific input and output modules I refer to these as “module” designs. There are a range of toolkits that fall into this module design that were designed for general purpose applications, and other designed specifically for music.

2.3.1 General Purpose Toolkits

General purpose toolkits created for interaction design enable fast prototyping and provide flexibility to support a variety of applications. Each of these toolkits provide software libraries to accompany their hardware modules and support serial communication to get sensor data into a computer for mapping to sound.

General purpose toolkits include Phidgets (Greenberg, 2001), Grove[15], .Net Gadgeteer[16], and Tinkerkit (Figure 2.12). These toolkits use keyed connectors that only allow legal connections. The software libraries make it easy to begin interpreting sensor data from individual modules.

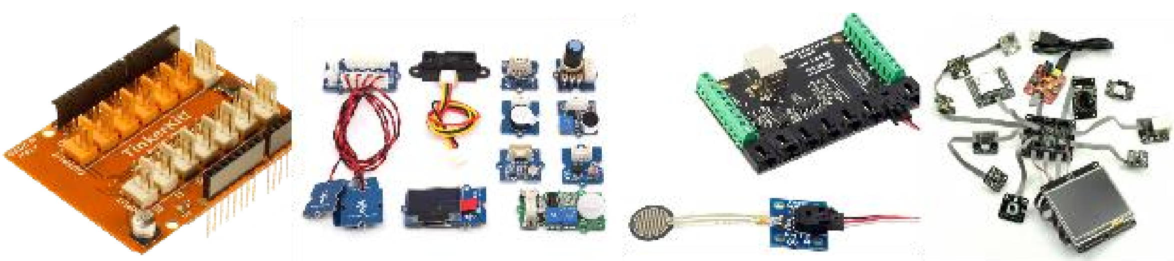

Figure 2.12 - General purpose module kits: Tinkerkit, Grove, Phidgets, and .Net Gadgeteer (from left)

This approach makes for a flexible toolkit, however, as I learned in the development of my own general purpose toolkit (see Chapter 3), the use of prefabricated cables, bulky connectors, and PCBs can limit the ability to embed sensors in musical interfaces. While premade I/O modules make it quick to experiment with different input options, this approach limits adaptability to other ways of working. This prohibits using craft

[15] http://www.seeedstudio.com/wiki/Grove_System
[16] http://www.netmf.com/gadgeteer/

approaches and malleable materials such as conductive paint, copper tape and other techniques available with a *breakout* approach and popular today in *DIY electronics* and *maker* communities.

For sound actuation, these toolkits all offer a motor driver module to drive one or two motors. Of these, only the Seeed Studio Grove and Phidgets support an external power source to support high power devices often needed for sound actuation and none support driving more than two motors. By contrast, the snare drum made by Raes for Aphex Twin drives sixteen different solenoids to provide a range of timbres.

The use of shields is another popular approach used today to scaffold the design and use of circuits. I include these in my description of *module* designs since they both provide functionality without the need to understand the circuit, they can be interchanged, and they are limited to the available shields (modules).

2.3.2 Tangible Interactive Music Toolkits

The toolkits designed for music can be configured to generate MIDI messages from sensor inputs, permitting control of software instruments without any programming or circuit building required. This allows musicians to work within the existing ecosystem of music software.

The I-CubeX system[17], Livid Instruments' Builder Kits[18], and the Teenage Engineering OpLab[19] all support a range of input modules. Only the I-CubeX system also support outputs, which can drive only 30mA, appropriate for an LED but not enough to drive a motor. I-CubeX offers software applications to route MIDI messages as well as C++ APIs to communicate directly with the hardware. Livid Instruments' Builder Kits can read

[17] http://infusionsystems.com/
[18] http://lividinstruments.com/products/builder/
[19] https://www.teenageengineering.com/products/oplab

in as many as 64 analog inputs and 128 digital inputs, and will register as a USB MIDI device. The Teenage Engineering's OpLab can read in 2 analog inputs to be used as MIDI controls, and offer a small set of sensor modules for exploring alternate forms of interaction. These platforms all provide an approach to creating custom interfaces using a range of analog and digital sensors to build a custom interface. MIDI support scaffolds mapping sensors to software. However, these closed systems only work with their own modules, preventing adaptability with other tools and approaches. Contrasted with the open 'breakout' approach, these tools do not invite the same versatility of applications and resulting designs. However, support for the MIDI protocol allows them to work within the existing ecosystem of music software.

2.3.3 Other Tools for Interactive Music

Outside of these hardware toolkits there are other tools for interactive music that scaffold otherwise difficult tasks. These vary in approach, enabling differing degrees of versatility, and adaptivity. As with the *module based toolkits*, these novel tools support some aspect of designing for interactive music but vary in scope of what they scaffold and how they integrate with other ways of working.

The Moog Music Werkstatt-01[20] (figure 2.13) is an analog synthesizer featuring a row of inputs that can be used to control parameters with custom circuits and an Arduino with a *breakout* approach (using a breadboard, jumper wires and electronic components). An Arduino software library supports writing programs to control the synth. The Werkstatt scaffolds the sound synthesis and provides a flexible platform for experimenting with different hardware using raw electronic components, a breadboard and jumper wires. This platform makes it easy to create novel interfaces to control the Moog synthesizer

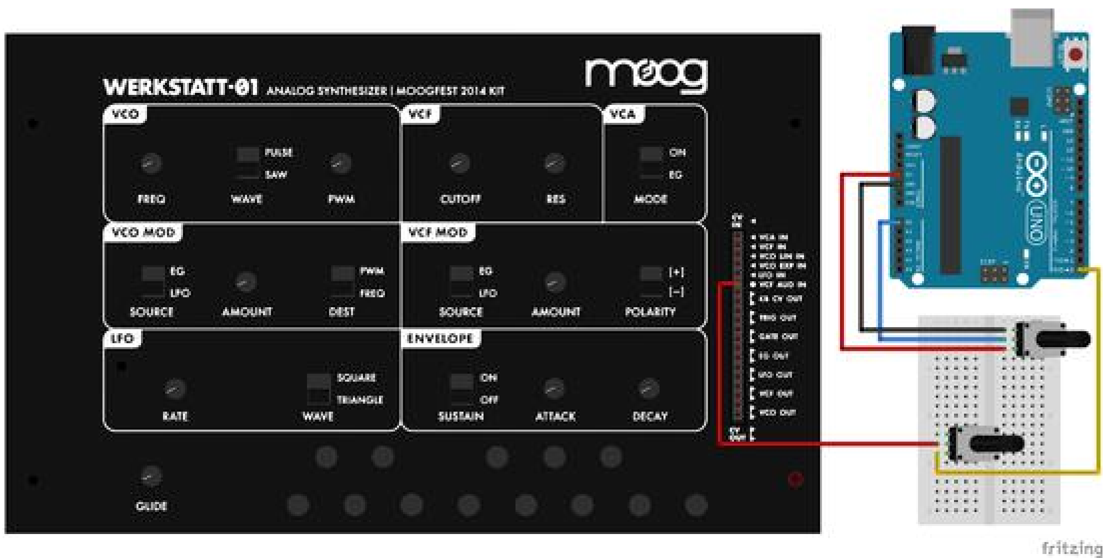

Figure 2.13 - The Moog Werkstatt-01 synthesizer. Moog provides a library for control with an Arduino

using the breakout approach, common in maker communities.

To lower the barrier to sound design in the visual programming language Max/MSP, Cycling ‘74 introduced BEAP[21], a collection of objects that scaffold sound synthesis using what they call the “analog model”. This is a way of conceptualizing sound synthesis that uses analog modular synthesis as a metaphor (Figure 2.14). This approach has similarities to what I have created for Pd with my Modular-Muse library. The scaffolding that BEAP provides lowers the barrier to sound design in Max/MSP. It

[20] https://www.moogmusic.com/products/semi-modular/werkstatt-01-moogfest-2014-kit
[21] https://cycling74.com/wiki/index.php?title=BEAP_Analog_Model_Curriculum_Outline

does this in a way that mirrors the existing ways of working in Max/MSP. BEAP is a versatile design tool that enables a lot of configurations, although is not completely open-ended as it is restricted to their “analog model”. However, since it is situated within the Max/MSP environment it can be used in conjunction with other objects in the Max environment, to make an adaptable and flexible tool.

Figure 2.14 – BEAP abstractions in Max/MSP

Two tools known as Maxuino and Pduino are designed to make it easy to connect an Arduino to Max/MSP and Pd respectively. These tools handle the tricky setup of serial communication with an Arduino with a system that can be configured from Max/MSP or Pd, without the need to do any Arduino programming. These tools scaffold the communication protocol, but they require some expertise to configure the ports as needed and to build working circuits. They provide flexibility in adapting to an open hardware design at the cost of complexity.

Another new addition to this space is the *Connection Kit* - Max for Live pack[22]. This is a software layer that simplifies using an Arduino within the digital audio workstation Ableton Live. The Connection Kit simplifies the process of mapping Arduino inputs to sound parameters in Ableton Live. This Connection Kit also provides connection LEGO Mindstorms, and LittleBits. The support for these connections provide several points of entry for interfacing with hardware that includes the breakout approach and two toolkits which are use a module approach.

The Maxuino, Pduino and the Connection Kit simplify the process of getting sensors from an Arduino into their respective sound environments. These tools support communication between hardware and software while the user builds circuits and maps inputs to a sound design patch.

2.3.4 Tools for Sound Actuation

In the realm of sound actuation the options are much more limited. While Pduino and Maxuino can be used to drive motors, there are few systems that make this easy. One example I have found is the BASTL Solenoid module[23]. BASTL makes units for analog modular synthesizers (the model used by BEAP) that also include a servo output module and sensor input modules. Their Solenoid module has four outputs that can be triggered using a “control voltage” generated by another module. On the output side, their module scaffolds the electronics needed and provides a way of triggering the four solenoid outputs, powered by an external power supply as with my MMMD boards. The user is left to shop for solenoids, and design a way to mount them (Figure 2.15). This approach from BASTL is accessible to modular synth users but requires mechanical design. The

[22] https://www.ableton.com/en/packs/connection-kit/
[23] http://www.bastl-instruments.com/modular/solenoid/

use of a control voltage paradigm works well for communities that use this type of gear but it limits compatibility with more typical ways of working.

Figure 2.15 - Different solenoid designs: push solenoid; rotary solenoid; pull solenoid (from left)

2.4 Designing for Between

Computational and interactive technologies are increasingly being used to enable new musical experiences; however, mainstream approaches to sound design and musical control do not provide musicians with access to the full creative potential of these technologies. There is a large gap between the low level circuits and programming used to build digital music instruments or musical robots and the consumer level tools typically used by artists and musicians. In the rest of this dissertation I describe how my work addresses some of these gaps.

I created a visual summary of the prevailing approaches for the three tasks of designing digital music instruments, interface design, sound design and mapping. These are framed by my design criteria in Figures 2.16, 2.17, and 2.18. The approaches are represented along the dimensions of support, versatility and compatibility. Support is indicated by the target audience, beginner or expert. Versatility and compatibility range from low to high.

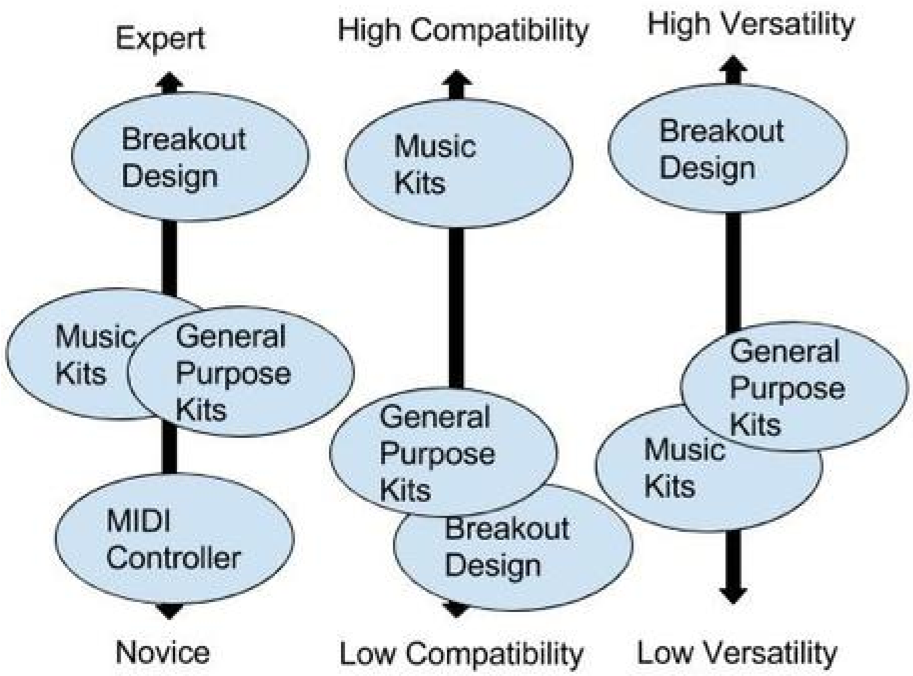

Figure 2.16 - A comparison of different approaches to interface design

In the field of new interfaces for musical expression the breakout design is often the preferred approach. This approach, while the most versatile, also has the steepest learning curve. Meanwhile music kits with MIDI support can be easily used to map to existing software instruments. This way of working is most compatible with techniques likely already familiar to musicians. However, the versatility of music kits is the lowest because they are limited to existing modules.

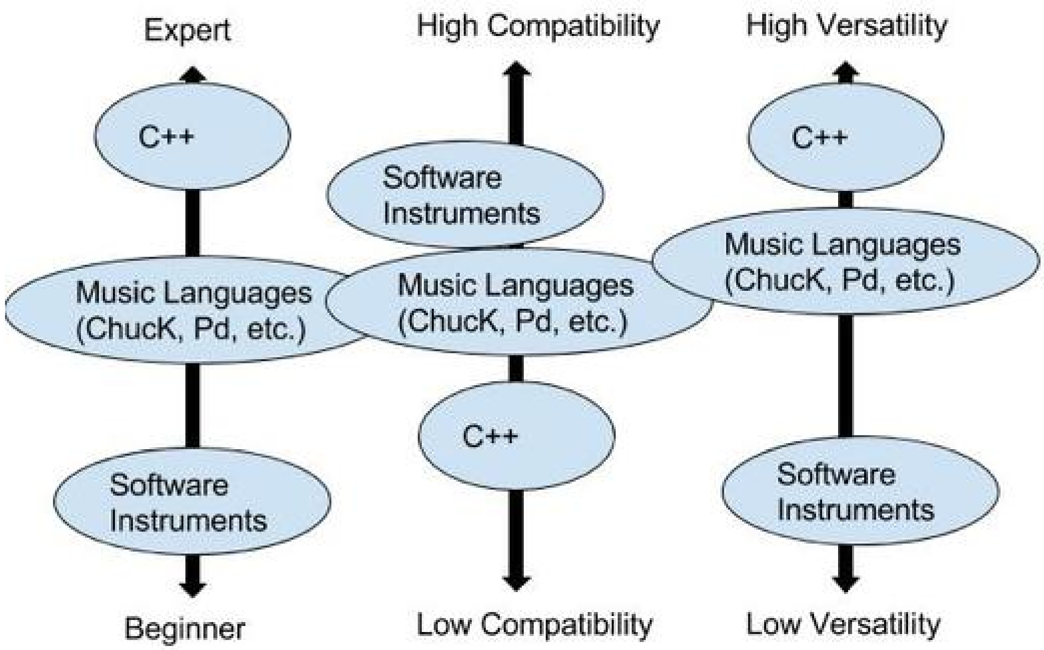

Figure 2.17 - A comparison of different approaches to sound design

As with interface design, options for sound design at the expert end of the spectrum also offer the greatest versatility but the lowest compatibility with ways of working familiar to

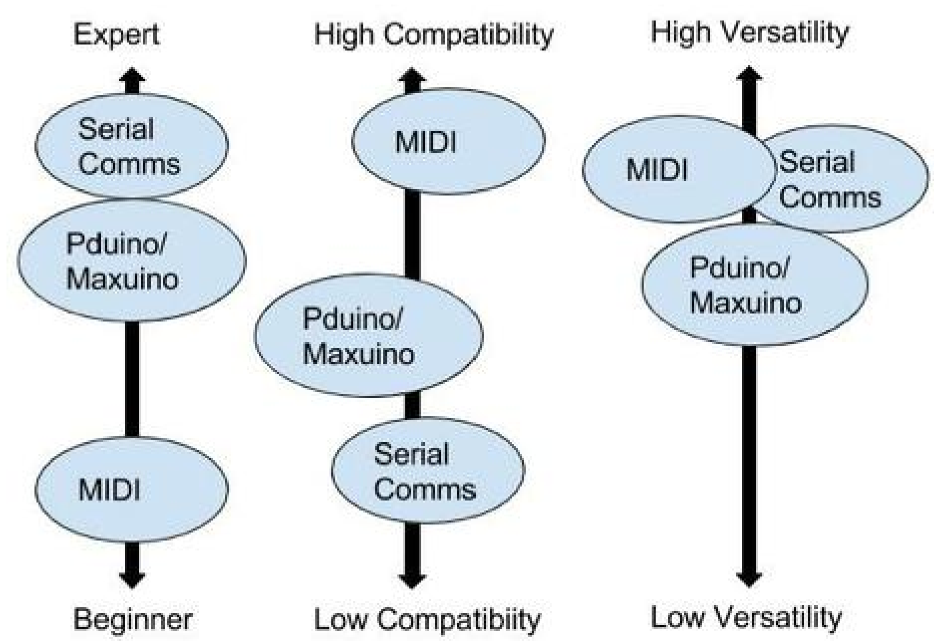

Figure 2.18 - A comparison of different approaches to communication and mapping

musicians. Software instruments are at the other end of the spectrum and musical programing languages sit in between. Although these languages do not have as much familiarity, they do tend to offer starting points that will mirror some aspects of software instruments or synthesizers, and they approach the level of versatility available with a low level language such as C++.

Mapping an interface to hardware involves a number of steps, but the focus here is on the protocol. MIDI is the easiest to use and the most familiar to musicians. It is also extremely versatile as it can be used with software instruments, audio programming languages and low level languages.

There is not a comparable ecosystem regarding musical robots. Outside of the BASTL modular synth solenoid driver, currently sound actuation must be undertaken with the breakout approach and building circuits, or using a motor shields and writing software to control those motors.

2.5 Summary

Research and art projects that use design for interactive music have a rich history that is nearly as old as computing itself. The development of interfaces to control sound emerged as computers were able to synthesize sound and respond to inputs in real time. Designing DMIs involves three main tasks: *interface design*, *sound design*, and *mapping* between the two.

The development of musical robots and other explorations in sound actuation have been pioneered by a small group of artists and researchers. In recent years there has been increased visibility of this approach to music with projects by established artists.

However, current approaches to sound actuation are severely limited and mostly require a great deal of electronics, programming and manufacturing skills.

From the literature on building creativity support tools, educational toolkits and constructionist learning environments, I draw three criteria for creating *scaffolded design tools for interactive music*: tools should be supportive, versatile and compatible.

The scaffolded design tools presented in this dissertation lower the barrier to working with interactive music technologies and illuminate alternative means of musical expression—specifically through designing DMIs, and musical robots. This work bridges the gap between ways of working familiar to musicians, and those familiar to interaction designers.

The next chapter details my foundational work that led to the development of my Sound Clippys and MIDI Bots platforms.

Chapter 3 Foundational Work

3.1 Example Projects

My first experiences working with interactive technologies was through the the design of digital music instruments (DMIs) and interactive art projects that use sensors, sound design and sound actuation. I will describe three projects made by wiring sensor, and motor driver circuits to an Arduino microcontroller and computer to interpret inputs and produce outputs. These projects represent a range of activities and the type of work my platforms aim to support. Next, I will describe my early attempts at building a toolkit to support this work and subsequent projects these led to.

3.1.1 Quadrofeelia

The first DMI I created, named Quadrofeelia (Figure 3.1), was inspired by the performance capabilities of a pedal steel guitar (Harriman et al, 2011). A pedal steel guitar uses pedals and knee levers that bend individual strings to raise or lower the pitch. Our design is laid out like a guitar with a digital fretboard for the left hand that acts as a slide bar to change the root pitch of the “strings” under the right hand. The four individual string controls allow each note to be bent up and down through four musical half steps, each based on the placement of the fingers. Unlike a traditional string instrument, the tuning of the instrument can be dynamically changed with the push of a button. A pressure sensor under the palm of the right hand allows the guitar technique of palm muting to dampen the plucked string response. Quadrofeelia uses the

Beagleboard[24] mini-linux computer, making it a self contained instrument. Because the computer could be embedded and dedicated to this instrument, the instrument is not susceptible issues that can arise when a personal laptop source of sound production. Most computers are subjected to operating system upgrades or changes to other software dependencies that may alter or disrupt the sound design programming for a digital musical instrument.

Quadrofeelia uses four digital inputs to select the tuning and seven analog inputs to play

Figure 3.1 – Quadrofeelia: inspired by the lap steel guitar (left); Playing Quadrofeelia (right)

it. This modest number of inputs took hours to solder together. This may be appropriate for a graduate level interaction design class but is prohibitive for beginners learning circuits.

3.1.2 Cacophonograph

In the Winter of 2011 I took a class co-taught by sound artist and inventor, Trimpin on interaction design for music at Stanford. I learned about mechanical sound actuation techniques through his work and with his advice created a sound art installation called

[24] https://beagleboard.org/

Cacophonograph (Figure 3.2). The music from an old schoolroom record player are interpreted by the robotic one-man-band. Audio from the record player is analyzed and used to trigger the home-made instruments. Cacophonograph uses rotary solenoids and linear solenoids to strum a cigar box guitar, scrape a washboard and to beat on a box drum, known as a cajon.

Figure 3.2 - Cacophonograph: installed at Modulations in 2011 (left); Scratching a record to play the sculpture (right)

While Quadrofeelia introduced me to the world of interactive sensors to control sound, Cacophonograph introduced me to approaches to sound actuation. This project was also an opportunity to explore the idea of art that is living, creative work that the audience can take in and interact with. Part of this is creating something both approachable and interesting for the audience. This initial experience with sound actuation was compelling, and I had a sense it was for others, too, who interacted with Cacophonograph. This project inspired me to continue exploring the opportunities and methods for sound actuation.

3.1.3 Calculated Captivations

Another experiment with sound actuation inspired the creation of my Modular-Muse platforms; an interactive sound art installation known as Calculated Captivations. Created with collaborator Kyleigh Morgan as part of the University of Colorado course

New Directions in Digital Art, this installation was displayed at the Boulder Museum of Contemporary Art in October 2012. Calculated Captivations featured four interactive "creatures" that responded to different stimuli including light, motion, and sound, with physical movement and sound actuation (figure 3.3). Each creature resided in its own terrarium on separate pedestals so they could be interacted with individually. This project further explores the possibilities of the basic motor driver circuit I used in

Figure 3.3 - Calculated Captivations, a series of four interactive sound actuated creatures

Cacophonograph to explore new forms of interaction and actuated responses that used vibration motors, and different types of solenoids and mechanisms.

For each of these projects I spent hours prototyping circuits with breadboards, and hand wiring and soldering to build sensor and driver circuits. Working with the *breakout* approach to electronics, in conjunction with audio software or programming languages, is a common way to design for interactive music. I used visual programming languages Pd and Max/MSP for these projects.

3.2 Teaching Interactive Music with Existing Tools

My interest in using these technologies inspired the creation of these boards but it was when I began teaching with these technologies that I saw another benefit and opportunity. My experience matches that reported by Blikstein and Sikitipiat, in that many students struggle with understanding and translating circuit diagram to physical components and connections, and how to integrate the use of a breadboard (2011). And of course, learning how to communicate with an Arduino to read and interpret inputs, trigger outputs, and create mappings, etc. presents its own challenges.

In teaching to undergraduates and adults I've seen how students resist changing circuits to try something different because they are afraid they won't be able to get it working again. In 2013 I taught a workshop on Pure Data and Arduino at Sparkfun. The workshop used the Sparkfun RedBoard (an Arduino clone), a breadboard and a handful of raw components[25]. In the six hour workshop adult participants built circuits for inputs (buttons, potentiometers and accelerometers) and outputs (LEDs and a DC motor driver for a solenoid). In this way it was very similar to typical approaches to teaching physical computing. This is what Blikstein and Sipitakiat refer to as the "breakout" model. However, instead of learning how to program the Arduino or another text based language such as Processing, this class used Pd to develop sound interactions and to trigger the solenoids with rhythmic patterns. It was a lot to cover, even in a full day workshop with adults who were eager to expand their knowledge of technology.

[25] Components used for the workshop can be seen here: https://www.sparkfun.com/wish_lists/75666

In this workshop and in courses where I have taught programming and physical computing through ATLAS and the College of Music at the University of Colorado, with this same breakout model, I became aware of the skills, knowledge, and number of steps needed in order to be able to design and execute ideas for creative uses of technology. The search for a productive way forward led me to the research on creating scaffolded design tools for education discussed in chapter 2. I apply these concepts to scaffold some of the stumbling blocks encountered when working with the existing tools.

My first approach to scaffolding these ways of working was a toolkit I named modular-mus (Harriman 2014). This toolkit evolved as I put it to use and eventually transformed into the Modular-Muse platforms detailed in the following chapters. Here I want to share the lessons learned, productive findings, and limitations from this early prototype.

Figure 3.4 - The modular-mus black box

3.3 Toward a Platform for Building DMIs

My original modular toolkit included hardware and software for constructing novel interfaces for building digital music instruments. The high level goals of this toolkit are still present in my current platforms. The library consists of a collection of objects that aim to simplify designing sound, and to assist in mapping the hardware to control and manipulate sound. This library has continued to evolve in this basic form, whereas the approach to hardware has changed more significantly.

The hardware component was originally conceived of as an all purpose toolkit focused on interaction design for music, similar to the module based toolkits for music described in chapter 2. I wanted it to be flexible—with ports, easily reconfigured as inputs or outputs, supporting a range of sensors and actuators. The first prototype used a metal black box and used 1/8” TRS audio cables as connectors (Figure 3.4). I built a range of modules that connected with a single cable to the brain module.

3.3.1 Evolving Prototypes

To make the inner workings of the toolkit more apparent, I switched to a more raw design without enclosures and built enough boards to create five kits. As with my current Sound Clippys Boards, the main boards use an Arduino Nano[26] clone. While the current Sound Clippys Board has 12 digital inputs, 8 analog inputs and no outputs, these hand wired prototypes had digital inputs, analog inputs and pwm outputs—four each (Figure 3.5).

This version had a “brain” board for interfacing to different components and a series of self-contained input and output modules, compatible with my original prototype.

Input Modules

[26] https://www.arduino.cc/en/Guide/ArduinoNano

A collection of digital and analog inputs were made to use the same stereo patch cable interface. Digital inputs included buttons and switches. On the interface board, digital inputs are wired such that a short between the tip and sleeve of the patch cable changes the digital input state. This convention is also used in available foot switches for music such as those used as keyboard sustain pedals and guitar amp channel select, requiring only a standard ⅛” to ¼” adapter to use as an input module with this toolkit.

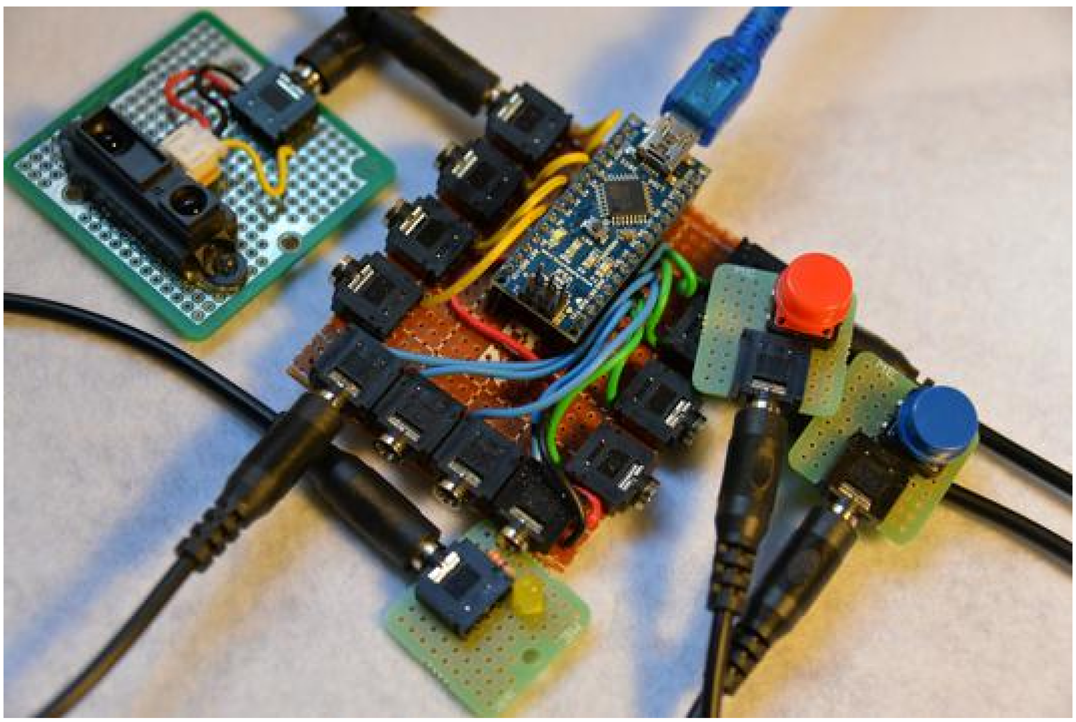

Figure 3.5 - Exposed version of my modular toolkit shown with input and output modules

Analog input modules include rotary and slide potentiometers, accelerometers, light sensors, membrane sensors, force sensitive resistors, and proximity sensors among others. One of the analog modules was a generic “voltage divider” module that allowed the use of variable resistors such as a light sensitive resistor, a bend sensor, or a force sensor. This flexible module became the basis for the variable resistor inputs on the Sound Clippys Boards (Figure 3.6).

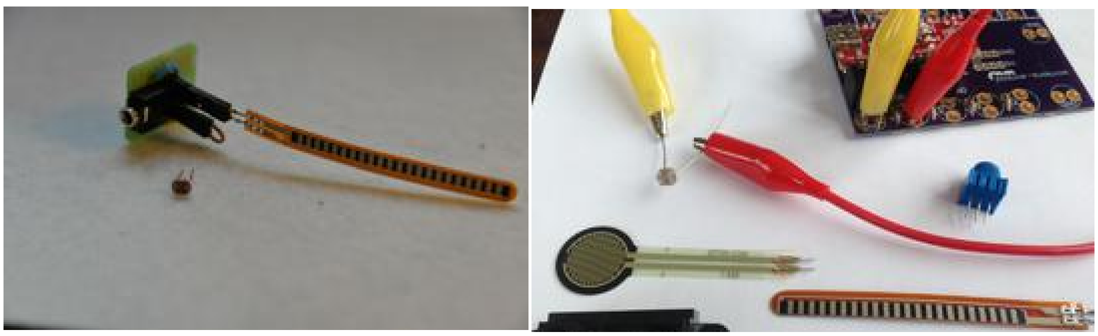

Figure 3.6 – Keeping what works: A voltage divider module for my original modular kit (left); the same circuit is used for the variable resistor inputs on the Sound Clippys Board (right)

Output Modules

This original toolkit design also had four dedicated output jacks wired to provide power and ground. Output modules include LEDs and a motor or solenoid driver board. This made it easy to use audio to control lights or to use other timing based triggers, which are convenient to program in Pd. The motor output module used a transistor for driving motors and solenoids. The USB powered Arduino 5V output that powers the circuit limits the size and number of motors that can be driven to a total of 500mA.

3.3.2 ideaLAB Workshop

In August of 2014 I taught a preliminary workshop with six 11-13 year olds at the ideaLAB, a public makerspace for teens at the Denver Public Library. The event, *Make*

Your Own Electronic Instrument, was offered as a free class and announced through the ideaLAB website and with flyers at the library, making for a self selected group of kids interested in the topic. Using my hand wired modular kits, participants learned the basics of synthesizing sound and controlling sound effects with hardware (Figure 3.7). During the 4 hour workshop participants built patches from scratch following my lead. Having the students build on their own is important to gaining proficiency, but in order to expose them to more ideas I found it necessary to change the balance so that not everything is built from scratch. In the evening between the two days of the workshop I built new example designs that could be used “out-of-the-box” to demonstrate concepts that we wouldn’t have gotten to otherwise. Seeing the benefit of a broad set of ready to explore examples, I have since further expanded the examples, along with extensive documentation in the library and on the project website, to make the objects and examples more self supporting.

Another significant change that resulted from this experience is on the hardware side. The original cabling caused issues if hardware modules were connected or disconnected

Figure 3.7 - Build your own instrument workshop at the Denver Public Library ideaLAB

while the board was powered up. Despite reminders, the students had a hard time remembering to disconnect the USB cable before swapping modules. This would cause the Arduino to reset, which would halt communications and in a few instances damaged the Arduino. The hardware needed to be more foolproof and robust to avoid a frustrating start to building digital musical instruments.

Another motivation for changing the design of the board grew out of what I saw as a significant drawback to the module approach. The participants were able to experience using a number of different sensors to map to sound, but the pre-made cables and bulky connectors were not conducive to embedding the sensors into objects.

Instead, my consideration of how to enable users to do “a lot with a little” (Resnick, 2005) led to a simplified and flexible approach. The Sound Clippys Board uses a handful of versatile circuits that can be used with raw electronic components, rather than pre-made modules. This approach strikes a balance with fast circuit construction using alligator clips, and circuits built into the hardware make it easy to get started. As I will show in chapter 5, this approach also makes it easier to integrate digital craft approaches and to embed sensors in a variety of ways.

3.3.3 My Prototypes in Interactive Music Projects

Having built the original toolkit prototypes for the ideaLab workshop, I quickly found other applications of this tool. I used it in my physical computing class to quickly demonstrate the way different sensors worked. Several students used parts of this kit to build projects. Meanwhile, I used it to create musical interfaces and other creative projects.

The first was a performance with the Boulder Laptop Orchestra (BLOrk). I developed an instrument and composition with George Ryan, an undergraduate student in the College

Figure 3.8 - *Picking Flowers* on display for the Communikey Media Festival 2014

of Music. *Lumelon*, was performed with BLOrk in the Fall of 2013. With light sensors plugged into my toolkit, eight performers used flashlights to control a series of musical loops.

I again used this toolkit to create an installation called *Picking Flowers*, shown at the Whether Weather pop-up art show for the Communikey Media Festival in 2014 (Figure 3.8). Each flower responds to changes in light with the sound of a virtual plucked string. It can be played by casting a shadow with your hand or in response to tree branches moving in the wind.

3.4 Sound Actuation

Cacophonograph and *Calculated Captivations* described above use sound actuation to produce sounds acoustically. These projects inspired the creation of the motor driver module for my modular-mus kit. However because the modules drew on the 5V power shared with the Arduino, and powered over USB, they could not drive more than 1 or 2 small motors. I created a new board that worked as a shield for the popular Arduino Uno.

I wanted to create a tool that would make it easy to drive a number of motors with more power. The motor shield PCB could be stacked with my modular-mus interface shield (Figure 3.9). This shield also supports an XBee Wireless module, allowing remote and wireless communication to trigger the motors.

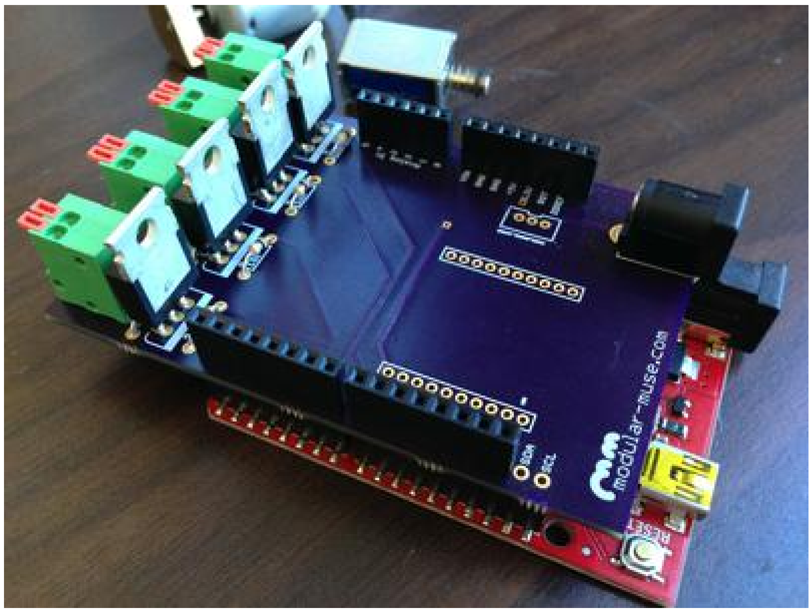

Figure 3.9 – Four channel motor shield used with the Kitsch-Instrument

3.4.1 The Kitsch-Instrument

Using this shield I developed the Kitsch-Instrument as a tool for exploring sound actuation (Harriman, Theodore, Gross 2015). The design goals were to support: *multiple points of entry* supporting a range of users needs and ideas, *transferability* of technique with existing systems, *transparency* letting users see how it's made, and *hackability* to enable variations and modifications.

Figure 3.10 - Solenoid attachments for the Kitsch-Instrument

The hardware allows exploration of different interactive sensors while the software library enables fast development of musical algorithms to trigger the solenoids used in the Kitsch-Instrument. The motor driver shield scaffolds the electronics to trigger the solenoids, but it is the mechanical design that makes the Kitsch-Instrument into a platform for exploring sound actuation by making it easy to attach actuators to different objects (Figure 3.10).

The Kitsch-Instrument employs three attachment mechanisms: A suction cup for smooth surfaces, a hanger attachment that can be hung from rims, and a cylindrical stand that allows solenoids to rest on horizontal surfaces. The various attachment mechanisms encourage investigation of different materials. The experiences with The Kitsch-Instrument in public showed it to be an engaging experience for young and old to explore sound actuation.

Since creating the Kitsch-Instrument I have iterated on attachment mechanisms for the same small 5V solenoids. My current designs use laser cutter making them much easier to reproduce than the handmade solenoid attachment created for the Kitsch-Instrument.

These laser cut designs were originally used in a series of workshops titled *Crafting Computational Percussion*.

3.4.2 Crafting Computational Percussion

Crafting Computational Percussion is a collaboration with fellow ATLAS PhD students Hyunjoo Oh and Abhishek Narula. We developed a kit comprising cardboard mechanical components and a custom printed circuit board, designed to support novices in building computational percussive instruments with everyday materials (Figure 3.11). The kit was used in workshops at NIME, and ISEA, and presented at TEI (Oh et al, 2016).

This kit is comprised of a series of cardboard mechanisms that we used to explore motion afforded by the mechanisms and the sound of everyday objects. This kit was designed using three considerations that closely mirror my criteria for *scaffolded design tools for interactive music*: accessibility, adaptability and expressivity. The kit was designed to be accessible to a wide audience, to be adaptable to use with different craft techniques, and to be an expressive medium that supports diverse results. The idea of adaptability relates to supporting the “improvisational materials” found with craft techniques.

Figure 3.11 - Mechanisms from the Crafting Computational Percussion project

To scaffold the activity we created the *Rhythm Board*, and a series of pre-cut parts to construct three different mechanisms. The *Rhythm Board* has connections for two servo motors and a solenoid, along with buttons, knobs, and analog sensors to control the motors. The hardware was pre-programed with a rhythm mode and a manual mode. This self-contained interface allowed users to focus on the activity of constructing the cardboard mechanisms and applying them in unique creations. The malleability of cardboard enabled easy modification and adaptability, resulting in diverse techniques and creations.

3.5 Summary

My experience building DMIs and working with sound actuation inspire the design space for my Sound Clippys and MMMD platforms. In teaching with existing tools I experienced the varied challenges of learning how to program, understand and build circuits, and make changes to and extend examples. However, when working with the toolkits, my original modular-mus, the Kitsch-Instrument, and Crafting Computational Percussion, I saw students and users get to creatively apply the technologies to create interactions, synthesize and control sounds, and to explore everyday objects for sound and construction.

The kits created for Crafting Computational Percussion and the Kitsch-Instrument both support designing with new technology, and embody the features of *Scaffolded Design Tools for Interactive Music* of being *supportive*, *compatible* and *versatile*. My current platforms build on these ideas to provide new supports and to enable greater adaptability with existing ways of working.

Chapter 4 Supporting the Design of Digital Music Instruments

There are three main components to designing digital musical instruments (DMIs). The interface design, the sound design, and the mapping between the two. Chapter 2 detailed common approaches for each of these tasks. The interface design often uses a “breakout” approach. “Module” based toolkits, including general purpose toolkits and music specific toolkits, provide a scaffolded approach to interface design. Sound design can be done with low level programming languages such as C++, with domain specific languages for audio, such as Max/MSP or Pd, or with pre-built software and analog

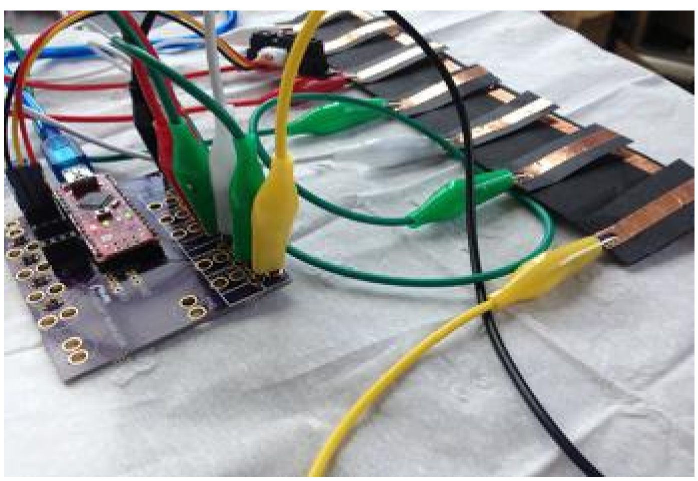

Figure 4.1 - The Sound Clippys board connected to a paper piano

instruments that can be used with external inputs. Finally, designers must create the connection between these, by mapping the inputs to the outputs. This depends on a *communication* protocol between the hardware interface to the sound design. Typical communication protocols include RS-232 Serial (via USB for modern machines) and MIDI. Finally data is parsed and mapped to a sound parameter or, as in more complex systems, the interface data is fed into an algorithm that can respond in different ways. This chapter details how my Sound Clippys platform was designed to support these activities, and my approach to teaching digital musical instrument design with this platform. This chapter also describes the contexts in which the Sound Clippys platform has been used.

4.1 Sound Clippys Platform Design

The Sound Clippys Platform consists of a hardware component for building tangible *interfaces* , the Modular-Muse Library for *sound design,* and a serial *communication* protocol connecting the hardware and software. The Modular-Muse library resides within Pd, the visual dataflow programming language for audio.

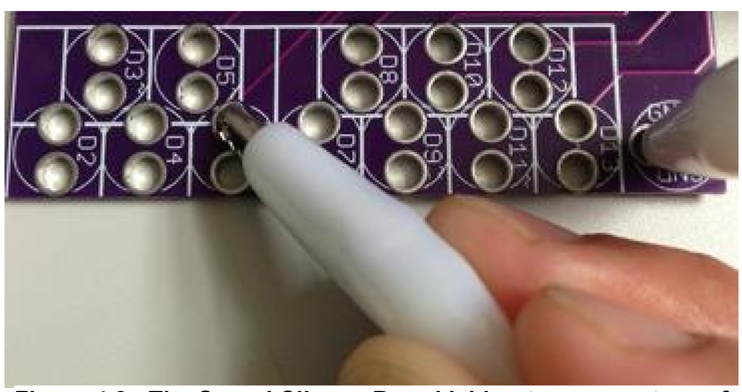

Figure 4.2 - The Sound Clippys Board laid out as one octave of a piano keyboard

4.1.1 Interface Design: The Sound Clippys Board

DMIs can be designed and built using the Sound Clippys board with simple circuits and readily available components and materials (Figure 4.1). The Arduino based board has 12 digital inputs (*buttons*) and 8 analog inputs (*handles*) that can trigger and manipulate sound parameters and audio effects. Circuits can be built using alligator clips with the Sound Clippys Board, without breadboards or soldering. The circuit connections are apparent—just follow the alligator clip to the destination.

My initial module based hardware described in chapter 3 used pre-built circuits and a single cable interface to connect a sensor or button. By contrast, the Sound Clippys Board moves away from the single cable interface and lets users build circuits with raw materials and electronic components. This approach removes a level of abstraction and scaffolds circuit building.

Both digital and analog inputs are made with simple circuits that are compatible with traditional circuits and digital craft techniques. The flexibility of the approaches supported by the Sound Clippys Board make for a versatile tool for building a wide range of musical instrument interfaces.

Digital Inputs

Digital inputs have two states, on and off. These can control a sound or the state of an audio effect, or be used to trigger events (e.g. sound sample playback) when an input changes. The digital inputs are along one side of the board with 12 alligator clip ports. The ports are laid out as 12 notes of a piano keyboard (Figure 4.2).

A digital input is toggled by making an electrical connection between digital input and the ground input. Figure 4.2 shows how you can play the 12 digital inputs with an alligator clip connected to one of the ground pins while the other end is touched to the digital inputs. The inputs can also be toggled by connecting the digital pins and ground through any conductive material (Figure 4.3). Using the Sound Clippys Board, makers have created digital inputs using wire, copper tape, conductive paint, and cello strings. The Sound Clippys Board uses a pull-up resistor on the digital pin to enable these simple digital inputs.

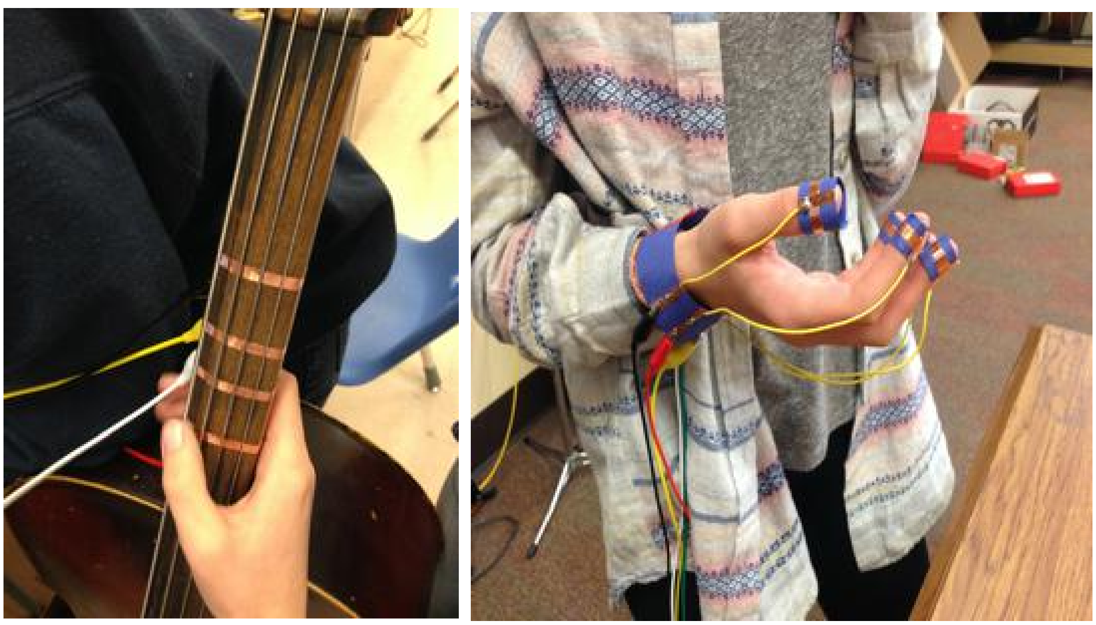

Figure 4.3 - Digital inputs made with the Sound Clippys platform and conductive materials: cello strings (left); copper tape and wire (right)

Analog Inputs

Analog inputs (*handles*) enable continuous, expressive control. The Sound Clippys board has two ways of creating analog inputs. Four of the eight analog inputs are routed to three pin headers for potentiometers and active sensors such as proximity sensors or accelerometers. The other four analog inputs are designed to use two clips that connect variable resistors including photocells, force sensitive resistors, bend sensors, etc. The Sound Clippys Board uses a voltage divider circuit with an onboard reference resistor to support the use of these sensors (Figure 4.5).

The variable resistor inputs can also be used for exploring the resistance of materials. For example graphite pencils can be used to create a variable resistor by scribbling on a

Figure 4.4 – Using graphite as a variable resistor, this drawing from a workshop at the library was used to control sound with Sound Clippys

piece of paper and touching two alligator clips to different parts of the paper (Figure 4.4). By moving them around you change the amount of graphite, and thus the resistance, between them.

For the boards used in my classes I soldered a small trimpot to one of the three-pin analog inputs. This adds a ready to use analog input to complement the single alligator clip approach for the digital inputs. This allows users to begin learning how to map analog inputs in software even before they have fully grasped how to build these circuits.

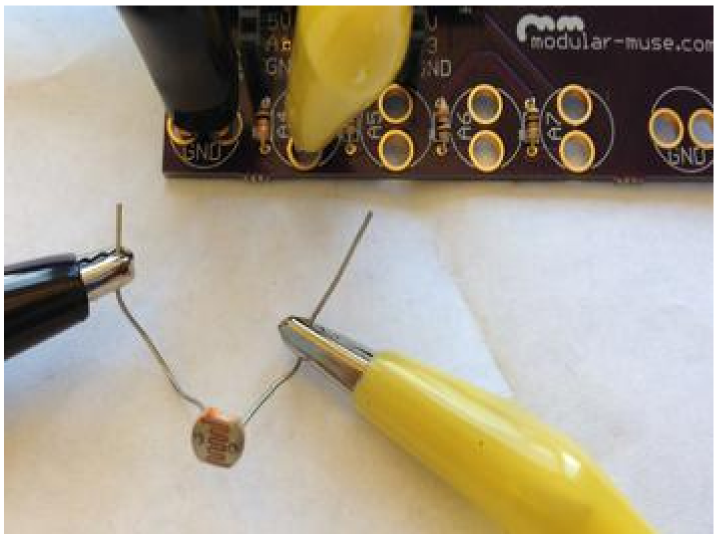

Figure 4.5 – Connecting a light sensitive resistor

4.1.2 Modular-Muse Library for Pd

The visual programming environment Pd uses a metaphor of patching an analog synthesizer, with on screen “wires” indicating connections and the flow of audio and control data. Pd is a visual environment allowing novices to trace the data flow of a program (known as a patch). However, to go beyond merely understanding the flow of a patch to modifying and creating new patches requires a significant commitment to learning about programming and the specifics of Pd. The Modular-Muse library for Pd lowers this barrier to entry, allowing even beginners to experiment with sound design patching in Pd.

As with the BEAP modules for Max/MSP (see Chapter 2), my approach focuses on the "analog model". This way of conceptualizing sound design matches with existing ways of working that draw on modular synthesizers and fit with the unit generator approach found in audio programming languages. The Modular-Muse library provides high level abstractions for sound synthesis, audio effects, and for reading inputs from the Sound Clippys Board. These objects can be combined in different ways using virtual wires in Pd. The Modular-Muse objects have GUI controls that can be manipulated with a mouse on screen, by other objects in Pd, or by mapping inputs from the Sound Clippys Board. The Modular-Muse library is built from existing Pd objects. This makes it easy to look inside to see how the library objects are built. This type of open, transparent scaffolding allows users to learn from the scaffolding and even make changes to it when ready.

The Modular-Muse library provides four categories of objects: Sound Generation Objects, Sound Control Objects, Audio Effects Objects, and Interface Objects. The library provides access to a wide range of sound design approaches and techniques suitable for beginners and adaptable to advanced uses in Pd.

Sound Generation Objects

The Modular-Muse library includes four objects for generating sound. The objects use classical synthesis techniques of waveform generation, pulse width modulation, frequency modulation and physical modeling using a plucked string model (Figure 4.6). For each sound generation object the first input sets the frequency (pitch) and the second controls the glide or portamento (the amount of time to change between different frequencies). The remaining inputs and GUI control other parameters specific to the sound synthesis technique.

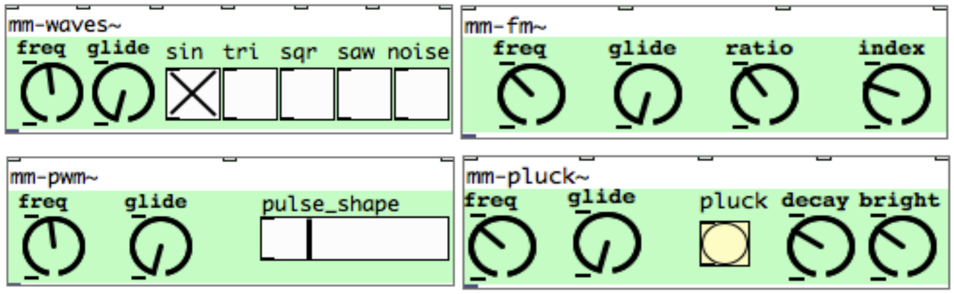

Figure 4.6 - The Modular-Muse library sound generation objects

Sound Control Objects

The Modular-Muse sound control objects are for shaping sound dynamically. These objects include envelope generators, a low frequency oscillator (LFO), and a sequencer for automating parameter changes over time. For example, an LFO can be used to add fluctuation to the frequency control of a sound generation object to create vibrato.

Audio Effect Objects

Audio effects change the character of sounds. The Modular-Muse audio effect objects include delay (echo), overdrive, reverb, low pass filter, and tremolo (Figure 4.7). These can be used with sounds synthesized in Pd, or with recorded and live audio streams from a microphone or instrument.

The first input on the Modular-Muse audio effect objects is the audio input, the second is an on/off control to use or bypass the effect. The dry/wet input controls how much the effect is applied to the original signal. The remaining parameters vary depending on the audio effect.

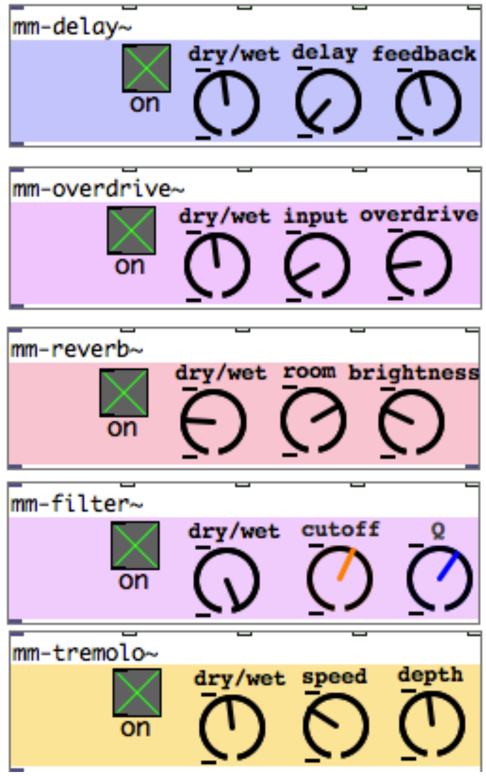

Figure 4.7 - The Modular-Muse library audio effect objects

Interface Objects

The objects in the sound generation, audio control, and audio effect categories provide support for sound design. The Modular-Muse interface objects support communication with the Sound Clippys board. These objects handle the serial communication with the Arduino and make it easy to route, scale, and map the inputs to sound parameters. The [mm] object connects to a serial port and parses the incoming data to route it to the

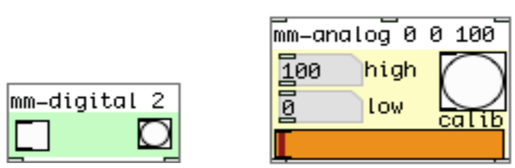

Figure 4.8 - The Modular-Muse library interface objects

analog and digital interface objects.

The GUI elements indicate the current state of the hardware (Figure 4.8). The mm-digital object has a toggle box that is empty when a digital input is open and an X when it is closed by connecting the input to ground. A second output produces a “bang” (a special event message in Pd) when the state changes from open to closed. The bang output can be used for triggering events.

The mm-analog object has a horizontal slider indicating the current value read at an analog input. The analog object might be used to control a frequency, an audio effect parameter, to read from an array of notes and more. The numeric ranges for these different sound parameters vary widely. To accommodate this, the mm-analog object can automatically scale the analog signal to a specified minimum and maximum value. The object automatically calibrates the analog inputs to the specified range based on the minimum and maximum value read at the specified input. A calibration button makes it easy to recalibrate after connecting a new analog input.

The Modular-Muse library provides a simple and versatile sound design toolbox. Combined with the Sound Clippys Board, it is a flexible platform for designing and exploring new interfaces for musical expression.

4.2 Teaching with The Sound Clippys Platform

In order to build DMIs with the Sound Clippys Platform users must at least know how to create inputs, and how to map these to sound parameters. I have used the Sound Clippys Platform in a number of workshops and classes, forming the case studies that I explore in Chapter 6. To illustrate my teaching approach, I describe development of the teaching progression I call *build-play-share-focus* that grew out of a project called *Pd*

Poems. I then describe how I introduce the Modular-Muse Library, the Sound Clippys hardware, building circuits with Sound Clippys, and basic DMI design concepts.

4.2.1 Teaching Approach

My approach to teaching is hands-on. I want to give students an opportunity to explore ideas on their own after being introduced to new concepts and tools. During the 2014-2015 school year I worked in a middle school music classroom as part of a project called eCSite. The goal of eCSite is to introduce *computational thinking* to students in middle and high school (grades 6-12) within the context of subjects that don't traditionally teach computing. I worked with a music teacher, Mr. Johnson, to create units that would convey concepts of computation for music.

In the first half of the school year I did not have the Sound Clippys Board. However we were able to install Pd-extended in the school's computer lab and use the Modular-Muse library to begin exploring sound design. This experience led to the development of the learning progression, *Build-Play-Share-Focus* (Harriman, 2015).

Build-Play-Share-Focus

The idea of the *build-play-share-focus* progression came directly from an experience teaching students how to use Pd and my Modular-Muse library. With students in groups of 3 or 4 I asked the students to build along with me to put together some basic patches (Build). After explaining how to use the generation, control, and effects objects together and where to find the help files and examples, the groups were given free rein to explore (Play). The results were chaotic and charming as the kids tried different parameters and combinations of objects. At the end of the class period the groups reported on their results. They described their sounds: “psycho-ambulance”, “electric-turtle”, “we made water". Groups were asked to present the sounds they created, the techniques they used, and to ask questions of their peer's (Share). The exploration is extremely valuable as the students gain comfort with the tools and explore ideas. They can explore the aspects they find interesting. The sharing session allows them to hear about different techniques and ideas they hadn't considered or settings they didn't have time to explore.

In the following class session Mr. Johnson and I talked about a task to ground their sound explorations (Focus). This is what came to be known as *Pd Poems*. Each group was given a poem and asked to design at least three sounds to accompany a reading of the poem. The poems chosen were rich with language relating to sound, describing rain, steam engines and rocks falling. When working towards a specific sound goal the students worked in a much different manner. They were recalling techniques they had explored when in "free-play" and applying them to the new task. There was a different energy in the room as teams focused and tried to analyze what changes to make to create their sound effects. During the next two classes students used the Modular-Muse library to design sounds to accompany their poems. The success of this activity resulted in the students performing their *Pd Poems* at the school's *Winter Arts Festival*.

In the second half of the school year I brought in the Sound Clippys Board. These experiences led to the development of activities and exercises I use to introduce the fundamentals of DMI design with the Sound Clippys Platform. These include sound design and mapping, creating digital and analog inputs, and interaction design for music.

4.2.2 Sound Design Examples and Using Hardware

To teach sound design with the Modular-Muse Library and Pd, I have the students follow along adding and removing objects connecting them with Pd wires. Once we know how to make sounds in this way the Modular-Muse examples can be explored. These provide a starting point to begin working with sound design concepts and to begin using the hardware and software together. The sound design examples demonstrate the basic function of the Modular-Muse Library, starting with sound generation objects, connecting them to control objects, and using audio effects objects.

Other examples demonstrate different mappings of the digital and analog inputs from the Sound Clippys Board. These demonstrate triggering audio samples, mapping analog inputs to control pitch and other parameters, and using digital inputs to turn notes on and off or to toggle audio effects. The main demo patch I use with the hardware maps the 12 digital inputs to a C scale to match the piano layout of the hardware. The on board knob on the Sound Clippys Board controls a filter effect. With this example running, the board can be played with a single alligator clip connected to ground. The other end can be dragged along the digital inputs while turning the knob to change the sound.

These example patches provide a starting point to represent some of the possible sound design techniques and mappings possible with the Modular Muse Library, while leaving open the interface design. After connecting the Sound Clippys board to an example, a student can begin considering how to build a hardware interface.

4.2.3 Buttons and handles

First, to help beginners understand how analog and digital inputs can be used to build a DMI, familiar objects provide powerful metaphors. Relating the concepts of digital and analog to features of familiar music instruments grounds these concepts in real world examples. For instance, valves on a trumpet provide discrete selection of pitch, while a trombone slide enables a continuous range. Using Verplank’s metaphor of buttons and handles, continuous control over an element of music is an example of a *handle,* an analog input, while discrete selection is an example of a *button*, a digital input. Inviting people to identify other examples of buttons and handles (musical and otherwise) provides an opportunity to create personal connections to these concepts.

Once students have a concept of what digital or analog inputs can be used for, I teach them how to build and connect the circuits needed for each.

4.2.4 Additional Materials

As described above, the Sound Clippys Board was designed to work with circuits built with alligator clips. Additional materials needed include a mix of electronic components, conductive materials and craft materials. The hardware design makes it compatible with wide ranging technologies and approaches. Here I describe some of the materials I have used in teaching with Sound Clippys, followed by the development of the *paper piano project*, an exercise I use to introduce making digital inputs.

Digital inputs can be built with conductive materials such as copper tape, conductive paint, conductive play-dough, and wire. In addition to these materials, cardboard, paper and found objects work well to create structure for instrument interfaces with simple tools: scissors, tape, glue, wire strippers, and cutters.

4.2.5 Building Paper Pianos

At the beginning of the DMI design unit at Manhattan Middle School I developed the *paper piano* exercise. This exercise introduces digital inputs and gives the students hands-on experience building switches. A step-by-step tutorial for building a paper piano is given on my website[27].

[27] http://modular-muse.com/?p=459

When I first introduced the Sound Clippys Platform in the school's computer lab, I demonstrated two examples. Both used copper tape to create switches, one with construction paper and the other with laser cut wood. I showed how to connect the Sound Clippys hardware to these interfaces and walked them through the function and mapping of my patch.

With a basic introduction to the hardware and our ongoing discussion about new musical interfaces, I thought the students had enough to begin creating with the platform. The students were given design journals and asked to begin developing project ideas, and begin working with materials. At this point, I imagined the students would begin identifying what construction materials, electronics and possible mappings they would use, detailing ideas in their design journals. Instead, they seemed stumped on how to get started. At the end of the second day of little activity it became clear they needed additional guidance. In order to create momentum towards making, I introduced the *paper piano* project.

In the next class period I demonstrated how to build a single switch using copper tape, construction paper and the Sound Clippys Board. I asked everyone to build a paper piano with at least two notes and encouraged them to consider different layouts than the examples I had demonstrated. Working with scissors, paper, and copper tape, the students built their paper pianos. Experimenting with different shapes, decoration, folds and orientations, the students produced a wide variety of paper pianos with these three materials and the Sound Clippys Platform (Figure 4.9). The results demonstrate the versatility of the basic materials and tools used in the paper piano project.

I have used this example in all of my workshops and classes to introduce the concept of digital inputs. I give a short demo and invite students to build their own working switch (build). Next, students are invited to experiment with different possible configurations (play). After exploring some initial possibilities, students are invited to share their creations and how they made them (share). The final step is the development of an

Figure 4.9 - Paper pianos created in the Music Focus class

instrument design of their own (focus). At this point they have hands-on experience with some key concepts of using the Sound Clippys platform.

Using Analog Inputs

The paper piano project conveys how to create digital inputs with conductive materials. I have not yet found a comparable exercise to introduce using analog inputs. One of the challenges is the variety of possible approaches.

There are two styles of analog inputs that correspond to a wide range of variable resistors and 3-wire sensors can be used with the Sound Clippys board. I have used proximity sensors and accelerometers and passive sensors like potentiometers for the 3-wire analog inputs. Variable resistors I have used include bend sensors, light sensitive resistors and force sensors.

For the case studies described in the next chapter I taught analog inputs by demonstrating first the knob on the board. Next I introduce variable resistors with one of several options and demonstrate how the different variable resistors work and how they can be interchanged with the Sound Clippys Board.

4.3 Summary

The Sound Clippys Platform was designed to lower the barrier to entry for designing DMIs. It consists of the Modular-Muse software library for Pd and the Arduino based Sound Clippys Board. The Sound Clippys Board uses simple circuits to create digital and analog inputs. The Modular-Muse library uses the *analog model* for sound design and provides interface objects to map inputs from the hardware to sound parameters. The alligator clip based circuits of the Sound Clippys Board can be created with a range of conductive and resistive materials. My approach to teaching with Sound Clippys developed while I worked with the platform for the first time in the *Music Focus* class. I have used this platform in classes, workshops and summer camps. The following chapter describes these experiences and the results.

Chapter 5 Sound Clippys Platform Results

I have used the Sound Clippys Platform to introduce DMI design to beginners. I describe the results of these experiences through four case studies. I describe the context including the participants, the teaching approach, the tools and materials available, and the resulting artifacts created using the platform. The artifacts have been analyzed to identify features of the Sound Clippys Platform utilized (or not), and the style of projects that have emerged. The categories I use to analyze the artifacts include: hardware inputs used, programming (none, some, or significant), and the instrument category, indicating the style of sound interaction produced.

In describing the results from these cases I highlight how my designs *supported* specific design challenges. I will show how they are *compatible* with existing tools and ways of working, and their *versatility* in how the platform is used and what can be created with it. It's important to note that the Sound Clippys Platform also precludes some uses and so I also discuss what *cannot* be created with it.

5.1 Modular-Muse Cases

I have used the Sound Clippys Platform in a variety of teaching situations with a diverse range of novice users. Each of these scenarios represents a unique case that I use to study the use of the Sound Clippys Platform in order to understand, what style of DMIs does the Sound Clippys Platform support in practice? What features are most commonly used? What is the range of outcomes produced with this platform? What materials and exercises are needed to facilitate success with this platform?

The cases include: Manhattan Middle School Music Focus, CU Science Discovery Research Experience, CU Science Discovery Sound Hackers summer camp, and a series of workshops at the Boulder Public Library called *Crafting Music Controllers*. The

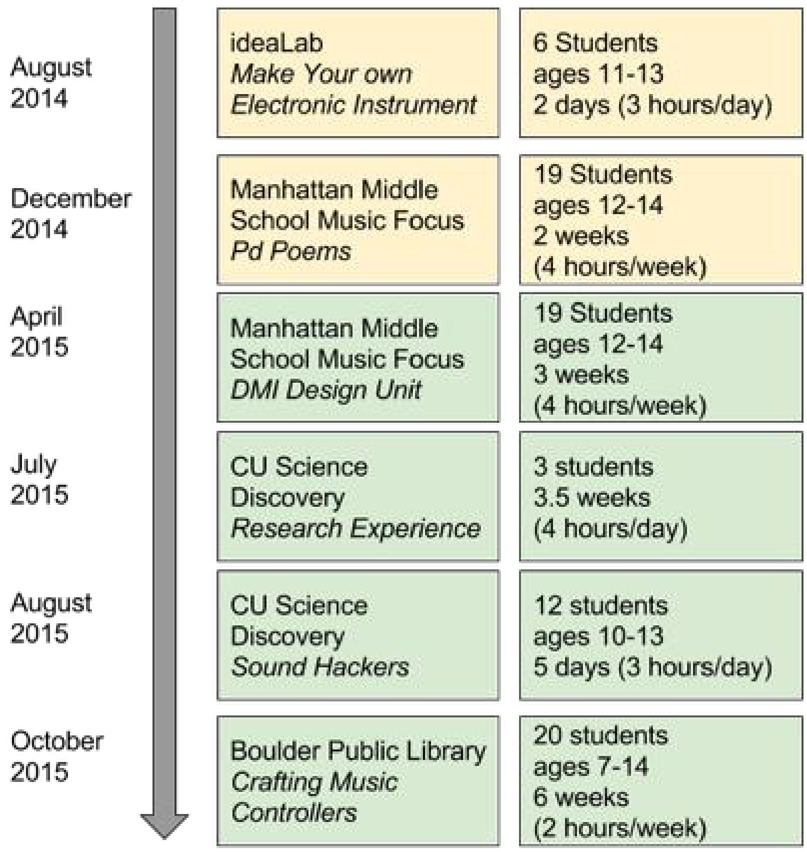

Figure 5.1 - Timeline of Sound Clippys case studies

diverse settings for these cases provide insight into a range of users and outcomes with this DMI platform. Figure 5.1 shows the timeline, audience and format for these cases. Preliminary work is indicated in yellow. These were described in Chapters 3 and 4. While the cases in green are described below.

I describe the context, my teaching approach and the resulting instruments and interfaces created for each case. These details show how the Sound Clippys Platform provides *support* to lower the barrier to entry for building DMIs and enable new populations to participate. The approaches described here exemplify *compatibility* with a range of technologies and existing ways of working. In particular, flexibility is demonstrated by the range of materials and techniques easily incorporated, all while working with an Arduino and Pd. Both tools are common in the NIME community. Finally, the artifacts created demonstrate the *versatility* of this platform.

In all of these cases I used a hands-on approach to teaching as described in chapter 4. Through hands-on exercises students learn about the hardware and software. These exercises included the *paper piano* project, an introduction to analog sensors and sound design with the Modular-Muse Library. Verplank's *handles* and *buttons* are used to explain the use of analog and digital inputs for building DMIs.

I first present the context and approach used for these different cases. Next I detail some of the instruments produced in each case study. A range of approaches and resulting instrument interface styles emerge. Below I detail the categories and the artifacts produced in these case studies and discuss the key findings.

The instrument/interface categories that emerged from these cases are *instrument inspired*, *augmented instrument*, *toy/game*, and *alternative interface*. These categories were identified by relating student instruments to categories from research on new interfaces for musical expression. The categories are based on gestural instrument interface categories defined by Miranda & Wanderley (2006). To this list I add *toy/game*. The *toy/game* interfaces could be categorized under the broad *alternative interface* category, however I wanted to acknowledge the strong theme of *play* that set these interfaces apart from other alternative interfaces. I define these categories as follows:

Instrument Inspired - Instrument Inspired interfaces draw on the physical layout and/or the function of an existing instrument, although they are not exact recreations of the inspiring instrument. These interfaces may look like an existing instrument but use different techniques to play them.

Augmented Instruments - Augmented instruments build on existing instruments. They can still be played as a traditional instrument but they add some new functionality.

Toy/Game - These are playful interfaces that make sound. They may be based on an existing toy or game and add some sound component, or they may introduce a new interaction that isn't necessarily musical, but makes sound through some playful interaction.

Alternative Interfaces - Alternative interfaces introduce new ways of controlling sound that does not mirror existing instruments. This category includes data sonification, interactive art works, and wearables.

I describe the results of these cases in chronological order. For *Manhattan* and *Sound Hackers*, I give examples from these instrument categories to demonstrate the diversity and show how these activities mirror categories explored by the NIME community. Meanwhile the results from *Crafting Music Controllers* and *Research Experience* represent different timeframes. Because *Crafting Music Controllers* was open each week there were students who spent as little as two hours with Sound Clippys. While for the *Research Experience*, students had 70 hours to work on their projects. The workshops for *Crafting Music Controllers* represent standalone activities for Sound Clippys while the instruments from the *Research Experience* case show how the platform can be used to create more sophisticated projects and provide a pathway to more complex Pd patches and circuits.

5.1.1 Manhattan Music Focus: Context and Approach

For the 2014-2015 school year I worked with Manhattan Middle School in a class called *Music Focus*. *Music Focus* is an elective—students are additionally enrolled in band, orchestra, or choir. The class had 19 seventh and eighth grade students ages 12-14.

I worked closely with the music teacher, Mr. Johnson, to integrate computation into the curriculum. Throughout the year I shared my own projects and we had ongoing discussions about computers and interactive technology as applied to music for composition, recording, arranging, sound design, and instrument design. The students initially used the Modular-Muse library during the *Pd Poems* project. Before building DMIs the class had an instrument design unit where they made acoustic instruments using everyday objects. My work with *Music Focus* led to both the build-play-share-focus teaching approach (Harriman, 2015) and the paper piano project that have been used throughout my teaching with the Sound Clippys Platform described in Chapter 5.

The unit on DMI design lasted for 3 weeks beginning in April 2015. This case was the first time I used the Sound Clippys Platform to teach DMI design.

In addition to the Sound Clippys Boards, construction materials, basic tools, and additional electronics were available. Electronics included solid core wire, copper tape and some analog sensors including IR proximity sensors, trimpots, light dependent resistors, flex sensors, and force sensors. Construction materials included construction paper, cardboard, tape, scissors and glue.

The *paper piano* exercise primed the students to begin their final projects. The simple paper interfaces they made represented a range of digital inputs they could now build. Since they shared their creations they were also introduced to alternatives created by their peers. In addition to the construction materials available in class I encouraged the class to consider how they might use other objects and materials in their everyday lives.

5.1.2 Manhattan Music Focus: Results

Over the next two weeks of class (about 4 hours each week) the class designed and built their instruments and interfaces. From the 17 projects created in the class a variety of interface categories emerged. The creations included instrument inspired interfaces, alternative interfaces, and those inspired by toys and games, and one augmented instrument. The results resemble historical examples in the development of new interfaces for musical expression, a field that developed out of computer music, and human computer interaction research. Here I describe some of the resulting instruments that came out of my work with *Music Focus*.

Music Focus: Instrument Inspired

Fifteen students created interfaces inspired by existing instruments. One was a guitar-like controller with a fretboard and a whammy bar made with a bend sensor (Figure 5.2). The left hand selected pitches by pressing down on switches made of cardboard and copper tape, another set of switches for the right (picking) hand triggers sounds, and the whammy bar detunes the pitches. This student had built a small guitar-like paper interface in the *paper piano* project. The paper piano became the prototype for his final project.

Another student made a basic piano-like layout with an additional switch to change octaves on the instrument. He crafted his interface with the approach used for the paper piano project, adding pressure sensors hidden under a layer of construction paper which he identified with marker. This student spent more time programming his instrument than creating the interface, building a patch from scratch to map the pressure sensors to control reverb and delay effects (Figure 5.8). He programmed the computer keyboard to enable and disable the effects.

Figure 5.2 - Guitar inspired controller

The *Paper Finger Drum Set* was inspired by a traditional drum set (Figure 5.3). It is a series of switches laid out on a picture of a drum set. The sounds for this instrument are samples of drums recorded in the music classroom. One of the examples was modified to add additional inputs and use these custom sounds.

Music Focus: Augmented Instruments

One student made an augmented cello. First she tested the conductivity of the cello strings by connecting the ground to the string and touching an alligator clip connected to one of the digital inputs to another part of the string. Having confirmed this worked she added copper tape to the fretboard under the strings. The copper tape was placed as if it were fingerboard tape which is used to mark fingering positions on string instruments (figure 4.2).

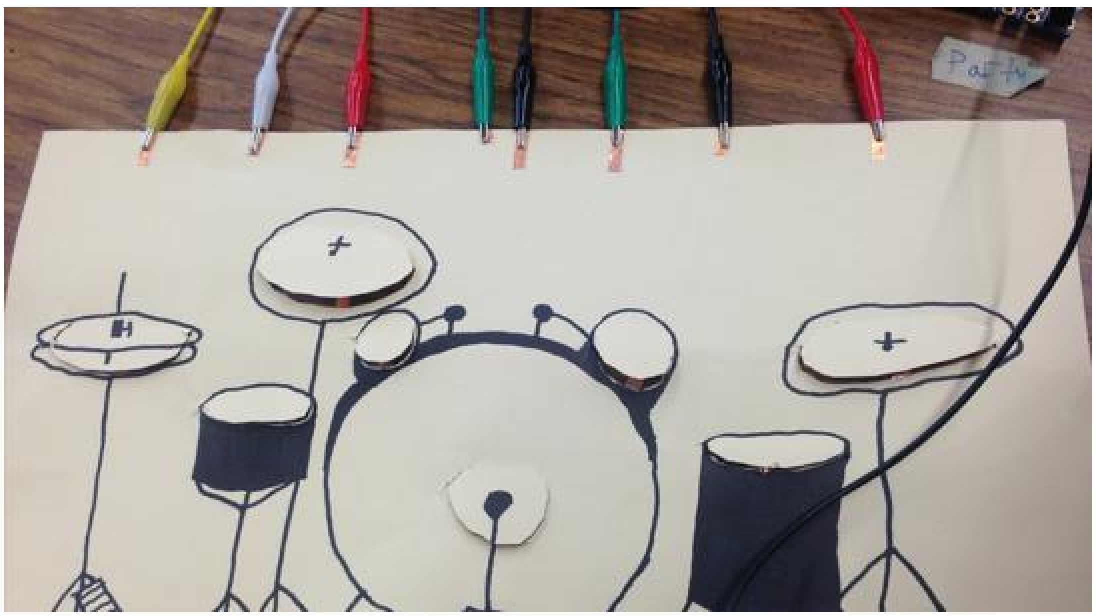

Figure 5.3 - The Paper Finger Drumset

When the strings are depressed over a fret marker the instrument produces additional notes. These additional notes can be played along with acoustic notes created by traditional means.

Music Focus: Toy/Game

Four *toy/game* interfaces were created in the *Music Focus* class. Of these two of them used existing toys (a Yo-Yo and a sock monkey stuffed animal), while the other two constructed their interfaces (a game of *Twister* and a light saber). The use of existing toys with sensors and the Sound Clippys board show how different sensors can be used to integrate everyday objects into interfaces. For the sock monkey the student attached bend sensors to the arms. This way when the stuffed animal was played with and manipulated, it would produce sounds. The student who used the Yo-Yo simply placed a distance sensor on the floor and played with the Yo-Yo above it. He spent his time programming a musical scale to select notes as the Yo-Yo went up and down. It was difficult to play but enabled an exciting concept, re-appropriating a toy as a computer input device. The lightsaber used a distance sensor placed on the end of a foam "sword." Care was taken in programming an FM synthesis patch to mimic the classic Star Wars light saber sound.

Music Focus: Alternative Interfaces

Instruments in the *Alternative* category introduce new sound and music interactions. In *Music Focus* this included four wearable projects, two interactive art projects, and one video game controller turned into an instrument. (The video game controller was categorized here and not the Toy/Game category because it was not intended to be a game, but an instrument played with the buttons on the controller). Of the students who made wearable projects, one made a glove with bend sensors on several fingers, programming each to control pitch based on the amount of bend. (Figure 5.14) Another glove based project created digital switches between the thumb (ground) and four fingers, enabling switches to be triggered with a snapping motion. Two students made shoe based interfaces—one built a switch directly onto the bottom of a shoe to trigger drum sounds, and another used the metal conductivity of tap shoes to trigger sounds on a special floor (a piece of cardboard with hot spots to play notes).

Other alternative interfaces from *Music Focus* included the "Musical Cloud", with electronics hidden inside of a cloud made of cotton balls. When the cloud was touched, LEDs blinked inside the cloud and the sound of thunder played. Another modified a painting, creating switches on the canvas so that it plays sound when you touch the painting. Cool colors triggered sounds of water and warm colors triggered the sound of fire.

Music Focus: Conclusions

These example instruments described show how students can pursue wide ranging projects. These project ideas grew naturally out of their interests. For example, the student who created the dance floor was a dancer, the student who created the guitar interface was a guitar player, the student who used the Yo-Yo was already carrying the Yo-Yo around in his backpack. The students displayed comfort in building the necessary circuits for their instruments and explored a wide range of interaction paradigms.

5.1.3 Science Discovery Research Experience: Context and Approach

The next group of students used the Sound Clippys Platform three months later, in July 2015. Three students ages 16-17 took part in a 4 week summer camp through CU Science Discovery called the *Research Experience*. Science Discovery is a long running program that hosts camps, science outreach programs and more. The Research Experience camp was new in 2015. Participants work in campus labs and present their work at the end of the camp. The three students who worked with me had requested my project based on a list of descriptions from campus labs participating in the program. They came to the ATLAS BTU Lab to work with me on the Sound Clippys Platform.

As part of the camp we had a budget of $100 to spend on additional components for their projects. In addition to spending time in the BTU Lab with me, the students took a research methods class twice a week with the rest of the students in the *Research Experience* (placed in different labs around campus). Through the research methods class they developed research posters related to their work in the lab and presented them at a public forum during the last week of camp.

As with the *Music Focus* students I gave them an overview of the Sound Clippys Platform, and led them through exercises to introduce the different components of the platform. After this initial introduction I asked them to work independently to research interactive projects and technologies, and to become familiar with Pd and Arduino. Additionally, they helped me assemble 25 new Sound Clippys Boards before working on their own projects.

After the first week they shared example projects they found inspiring from maker, and interactive art and music communities. I also shared example projects of my own. The next task was to develop their own project ideas. They browsed the Sparkfun and AdaFruit websites for ideas on sensors they were interested in using.

5.1.4 Science Discovery Research Experience: Results

The resulting projects exemplify the versatility of the Sound Clippys platform. Two students elected to make sonification projects (Figure 5.4). Sonification is the use of sound to convey data or information. Data can be mapped to control different sound parameters hear changes in the data through the sound. The sonification projects show an application for this platform beyond building DMIs. The third student created a glove based “air guitar” called the GuiSam (Figure 5.4). To give a sense of how my platform supported these creations, I will describe how these three instruments were constructed with the Sound Clippys Platform.

Gas Sensor Sonification System

The gas sensor sonification project used four different gas sensors to trigger sound files. The sensors used were Liquid Petroleum Gas (LPG), Carbon Monoxide (CO), alcohol and Hydrogen gas. These sensors work on 5V and ground to produce an analog output. A single resistor is used in the circuit in conjunction with the sensors. The sensors were wired into into the board's 3-pin analog inputs. In Pd the data was parsed into low, medium and high ranges to trigger playback of different sound files. The end result served as an alarm system that used sound to signal dangerous levels of the various gasses.

Weather Station Sonification

The Weather Station Sonification project plays different melodies based on the weather. The sensors—wind speed, temperature, humidity, rainfall and light— are mapped to their own unique sounds and play a melody based on the current conditions. This resulted in interesting conversations about what kind of melody might be played for heavy wind or for light rain, etc. The idea was to create a system to communicate the weather in a non-traditional way that doesn't require looking at a display. This could, for instance, provide a background ambient soundtrack that represents the current weather conditions.

Both sonification projects provided opportunity for the creators to consider the role of sound in human computer interaction. These projects also revealed an alternative application for Sound Clippys, as a tool to explore not just music, but alternative data representations.

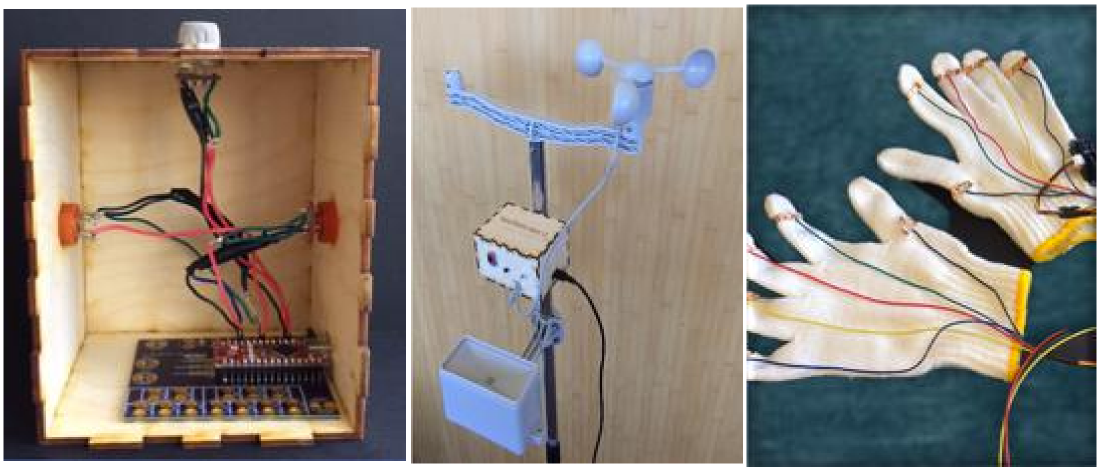

Figure 5.4 – Projects created during the Science Discovery *Research Experience*: Gas Sensor Sonification System; Musical Weather Station; GuiSam (from left)

GuiSam

The final project in the *Research Experience* group was the GuiSam—a wearable “air guitar”. The GuiSam uses four switches on each hand that are triggered by bringing fingers to the palm. As with a traditional guitar, the left hand selects chords, and the right hand triggered notes. A distance sensor allows the position of the left hand changes the root pitch. The Modular-Muse [mm-pluck~] sound generator and overdrive objects give it a convincing guitar sound. The result is a very playable, instrument inspired DMI.

5.1.5 Science Discovery Research Experience: Conclusions

The students in the Research Experience demonstrated how the Sound Clippys Platform can be used for invention and provide opportunity to explore wide ranging concepts at the intersection of interaction design and sound. The students developed their projects over several weeks as they became familiar with my Modular-Muse Library and the standard electronic components I use with the hardware board. The additional time spent learning about additional sensors and how to program with Pd allowed them to develop more sophisticated interactions and projects. The Sound Clippys Platform allowed them to focus on high level details like the interface and sound design, as well exploring different approaches to conveying data for the sonification projects and mapping chords and scales for *GuiSam*.

5.1.6 Sound Hackers: Context and Approach

In August 2015, twelve students ages 10-13 took part in the Science Discovery camp called *Sound Hackers*. The camp ran for 3 hours a day for five days in the BTU Lab. A five day curriculum was used with a series of exercises for exploring the different topics. I gave the students design notebooks to facilitate class exercises and to capture their design ideas. The first 3 days were dedicated to learning the basics of the platform. At the end of the third day students proposed their project ideas, describing what they wanted to build, how it would be used, and the sounds it would make. The final two days were used to work on their projects with an hour reserved to demonstrate their creations at the end.

5.1.7 Sound Hackers: Results

The structure of *Sound Hackers* did not allow students long to contemplate their designs or to acquire many materials; they had to make do with what was available in the lab and items they already had at home. After presenting their initial ideas, students were invited to form teams interested in working on the same project. Some of the resulting projects are described here. These examples show new variations in the instrument categories defined above.

Instrument Inspired

The Manual-Table is inspired by a record turntable. It uses a cardboard “disc” with a wire protruding out of the bottom to make contact with the surface below (Figure 5.5). This wire, connected to the ground port, comes into contact with eight individual strips of copper tape which were connected to different digital inputs, closing a switch depending on where the disc is in its rotation. This can be viewed as a variation on the paper piano but it also has a quality of composition, as the notes being triggered are always in the same sequence. When demonstrating their instrument, the inventors pointed out that the notes can be played forward or backward, as with scratching techniques used in turntablism.

A tunable drum pad was created using a snare drum head brought from home with a piezo vibration sensor and force sensor. The force sensor was used to control the playback rate of a snare drum sample. This mirrors a technique used with acoustic drums where the tension on the head is changed by pressing down with a palm or an elbow to control the fundamental pitch of the drum.

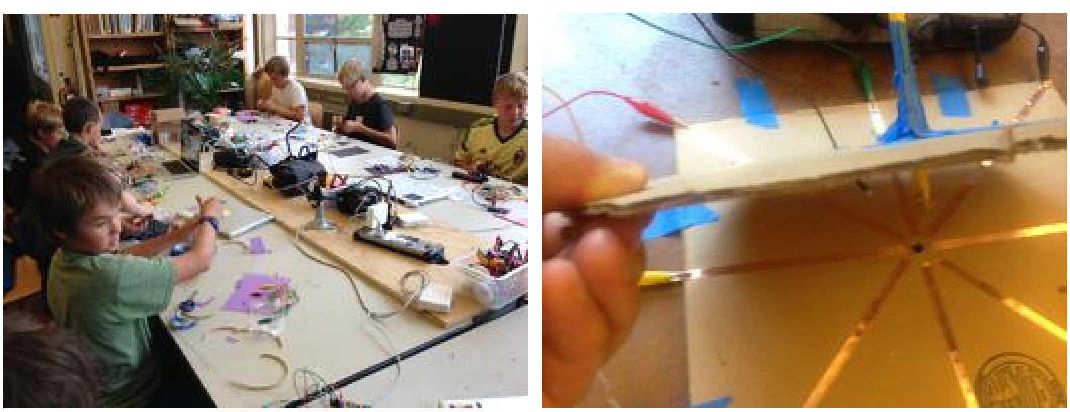

Figure 5.5 - The Sound Hackers workshop: constructing paper pianos (left) Showing the design of the Man-Table (right)

Augmented Instrument

One of the students in Sound Hackers elected to not build a physical interface beyond the Sound Clippys Board itself. He used alligator clips directly with the board to toggle effects on and off for his voice. Other digital inputs set effects to presets while two knobs tweaked parameters. This represents an approach that is well suited to situations where making physical interfaces is limited. It leaves a lot of room for learning the basics of programming with Modular-Muse and Pd while still using the hardware inputs.

Alternative Interfaces

One student wanted to create a digital air horn. When asked what inspired this idea he shared a version of Beethoven's 5th Symphony performed on air horns that he found on the web. He brought in an air horn from home and took a computer outside to record a sample. The air horn also became the interface, using copper tape to create a ground on the button on air horn. He then added copper tape to a glove to trigger the recording. The sample played back at different speeds to create different notes, depending on the finger pressed.

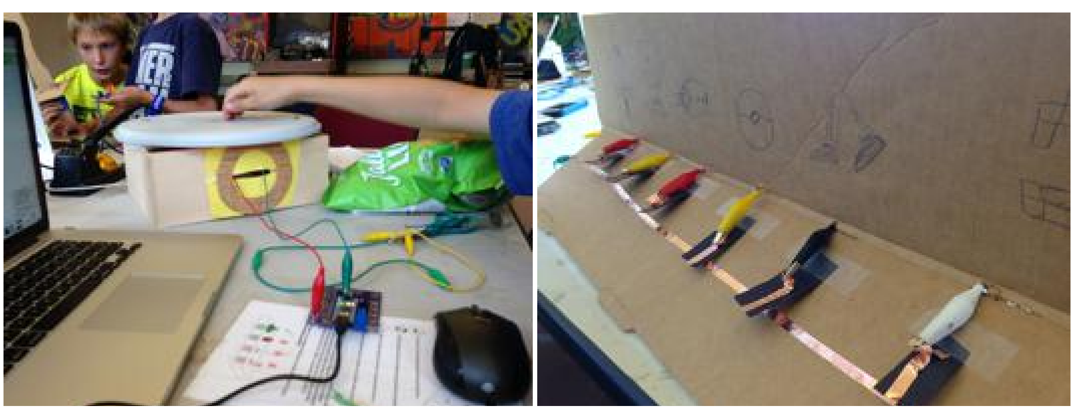

Figure 5.6 – Sound Hackers projects: Tunable digital drum pad (left); video game soundboard (right)

Toy/Game

Possibly inspired by the sword example I shared with them from the Music Focus group, two students created sword interfaces. These students worked together creating their individual swords using a distance sensor on the tip to control the pitch of a synthesized sound and a button on the handle to trigger a sample of a slashing sword. At the end of the camp they demonstrated together, with a sonified sword battle.

5.1.8 Sound Hackers: Conclusions

This camp demonstrates how the Sound Clippys Platform enables beginners to design DMIs with simple and understandable materials and activities. The hardware was quickly learned and used to produce diverse projects in only 5 days. However, the interfaces developed in this camp relied heavily on pre-made example patches from my library.
A lot of factors influence the amount of programming students are able to learn and apply including: understanding the flow of a patch, how to make changes or build new sound synthesis patches, even what they want to do. It may also be that given the choice, most students prefer constructing the physical object to the digital sound design. Following the description of the final platform case study, I discuss some of the challenges and tradeoffs of my approach and some alternatives that might influence sound design and programming.

5.1.9 Crafting Music Controllers

The final case I describe took place in the fall of 2015. I taught a series of six workshops at the Boulder Public Library called *Crafting Music Controllers*. These workshops were advertised through the library's website as part of their *STEAM* (Science, Technology, Engineering, Art, and Math) programming. Registration for the free classes was capped at 20 students, with the registration page stating the classes were for students ages 8-14. Each class lasted 2 hours. For each of the first four weeks I presented a different topic drawn from my curriculum for the Sound Hackers camp, while the final two weeks were billed as “open hack” sessions for students to work on projects or ideas that interested them. The workshops were advertised as a series, but allowed for students to sign up for one without committing to the whole series.

Because of the nature of the programming at the library it was necessary to treat each session as a standalone unit. Each week there were new students while others attended all six sessions. I created four standalone individual activities that I taught during the first four weeks: paper pianos, analog sensors inputs, sound design, and playing notes and rhythms. The structure limited most projects to a single session although some students worked on projects in between classes, or brought them back to continue working on them.

5.2 Discussion

These case studies show a range of settings where the Sound Clippys Platform has enabled beginners to build new musical interfaces. Here I show how these results demonstrate my design criteria of *support*, *compatibility* and *versatility.* I will describe how these criteria impact three components of DMI design: *interface design*, *sound design* and *mapping*. I show how mapping and sound design are closely related to the available example designs for these case studies. This is especially true for *Sound Hackers* and *Crafting Musical Controllers,* where students had limited time with the platform. I also show how the simple circuits support a broad range of interfaces and approaches, and describe challenges and areas for improvement to better support DMI design. These results demonstrate activities and artifacts that mirror those found in related artistic practices and research communities.

5.2.1 Approaches to Sound Design and Mapping

In these four cases students used Modular-Muse library in vastly different ways when doing sound design and mapping. Some students used my example patches without making any changes, while others created new patches from scratch. Those who used example patches without making software changes were instead focused on building circuits and a physical interface. These students still had decisions to make regarding sound design and mapping. Selecting an example determines the type of sounds and inputs available. Next they determine the hardware inputs used by the patch and decide which of those to create an interface for.

This is seen in *Music Focus* with the *Musical Shoes,* where an old pair of shoes were outfitted with switches. The shoes were connected to an example patch to trigger drum samples from the digital inputs. This approach is also seen in the *paper pianos* exercises

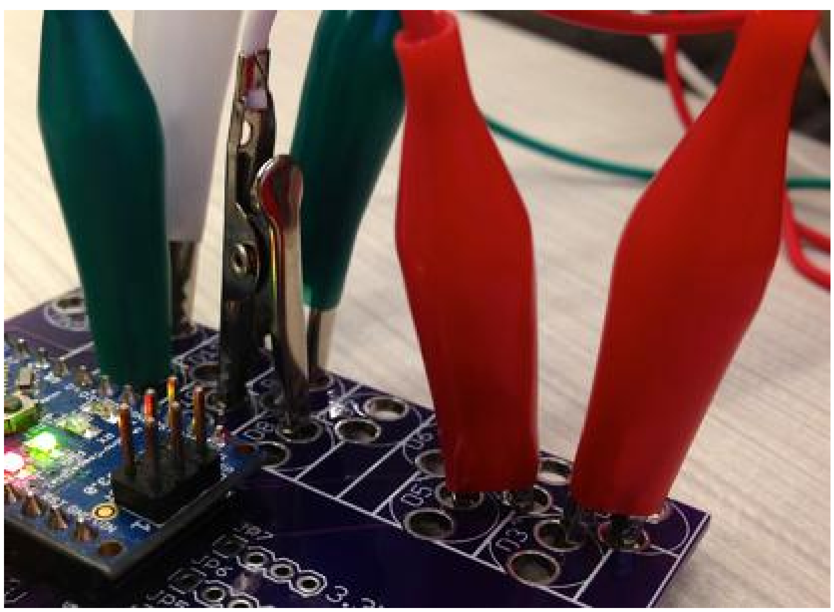

Figure 5.7 - Connecting a paper switch to more than one digital input

where the patch can play up to 12 different notes. Students can choose how many and which notes they want to trigger. They select the available notes of their instrument by what inputs they connect to. The tuning can be changed by changing the connections. One student in the *Crafting Music Controllers* workshop discovered he could trigger multiple notes with a single switch by connecting his alligator clips to create switches on more than on digital input at a time (Figure 5.7).

The students who take this approach have some control over the sound design and mapping options, but are also limited to the available example patches. Additional example patches would allow more variety with this approach. An online repository for sharing and downloading patches would increase the options for students who can't or don't want to program their own.

The next level of engagement with sound design and mapping involves making minor changes to example patches. If a student can reassign or reroute an analog or digital inputs in software, or if they can swap a sound generation object, an audio effect object or a sound sample, they can explore new sound designs and mappings.

Examples of this include the *Man-Table* and *Paper Finger Drum Set*. With the *Man-Table* the designers changed the mapping of a low pass filter to use one of the variable resistor inputs instead of the onboard knob. To make the *Paper Finger Drum Set* the designer started with an example patch that plays five samples and expanded it to support eight inputs. He then recorded instruments from the percussion closet at school and edited the samples and loaded them into his patch.

These students demonstrate control over mapping and sound design by making simple changes to example patches. Moving beyond this level of engagement with the tools only occurred in a few cases, as most students make only minor changes to example patches or none at all. The outliers were in the *Research Experience* camp and *Music Focus,* where students spent more time learning to program with Modular-Muse and Pd.

Their projects demonstrate more customization, and sophisticated mappings. For instance, one student in *Music Focus* programmed a patch to use keyboard keystrokes and multiple force sensors to toggle presets and control audio effects. His paper interface also had an input to change the notes in the four keys he made.

The sonification projects from the *Research Experience* camp demonstrate how the Modular-Muse library can bring in data from alternate physical sensors. The students developed algorithms in Pd to interpret the data according to the sensor type. Next they made determinations about how and when the data should be represented.

These student projects show comfort with Pd beyond making small modifications or duplications of example patches. These demonstrate that students can gain experience with a range of programming concepts in Pd.

However, in the *Sound Hackers* camp and *Crafting Musical Controllers* workshops no students built patches from scratch for their projects. Even in *Music Focus* where students had previous experience with the library, most started with an example patch as the basis for their project. Clearly there are some barriers that limit students from engaging with in-depth Pd programming. I witnessed a number of challenges beginners encounter when working with Pd and the Modular-Muse library. Some of these challenges are inherent in my current approach, while alternative solutions have different tradeoffs.

Programming and Sound Design Challenges

I argue above that students can engage with sound design and mapping by selecting an example and choosing how to connect to it. However, this approach is inherently limited. To realize the full potential of DMI design and more broadly, computation for music, requires a path to programming. My approach with the Modular-Muse library is to introduce concepts of sound design with high level building blocks in Pd. Some of the challenges beginners face have to do with foundational concepts in Pd and audio programming in general. Despite teaching students these concepts, I repeatedly observed students struggling with the following:

- Edit mode and Performance Mode
- Connecting objects
- Mapping input ranges
- Translating between notes and frequency

These challenges lead to confusion and frustration on the part of users and limit progress as students work to overcome these. This led to the large number of projects where little or no programming was used.

Edit Mode and Performance Mode

In Pd and similar environments there is no difference between the programming environment and the runtime environment. To do one or the other is a matter of edit mode vs. performance mode. Edit mode is used for programming or making changes to patches. To interact with a patch (to perform it) you must change to performance mode. Some users pick this up quickly while others struggle to understand when or how to change modes.

Connecting Objects

Another common challenge is knowing how objects should be connected. Often users fail to realize there are multiple inputs and outputs on objects and make connections between inlets and outlets (Pd object inputs and outputs) arbitrarily. They assume the program understands their intention. For instance a student might add a new audio effect that they want to map an analog input to control. They add the audio effect object and connect a sound generation object to it. Next they add an analog input object and connect it to the first or second inlet, the same one where their audio signal is connected. The analog input is working and they can see it moving but the sound is not being affected because the analog input has not be routed to a control input of the effect. Despite my effort to make clear the function of the inputs by visually aligning them with the GUI controls, many students fail to understand the object inputs.

Another challenge for beginners making patch connections is the difference between audio rate and control rate signals. The distinction is an important one for processing efficiency. Audio signals must be processed at the sampling rate, while control signals are processed at a fraction of that rate, typically based on the audio block size used. The use of audio rate and control rate signals is foundational in Pd and many other audio programming languages. However the concept is difficult for beginners with no previous programming experience, let alone audio programming.

Mapping input ranges

Even with knowledge of how the different inputs work, to effectively map a sensor to an audio parameter, users must understand appropriate number ranges for the given parameter. Otherwise, even with all the right connections there will be no discernable change in the sound. I repeatedly saw students connect an analog input to an effect parameter without specifying a range. Without a range specified the analog object uses the default range of 0 to 1. This is appropriate for some inputs but for others the parameter will be appear stuck at the low end of the range.

Translating between notes and frequency

The final challenge I discuss here of translating notes to frequency forces users from the realm of music to that of physics. Some students with musical training wanted to be able to play specific chords or notes to create a melody. This requires understanding both an abstraction of the notes and a conversion. Typically this is done by using MIDI note numbers to represent musical pitches. For example the number 60 represents middle C, while the number 61 represents a C# and so on. Middle C corresponds to a frequency of approximately 261 Hz while the C# above middle C has a frequency of 277 Hz. The Pd object [mtof] provides an easy translation between MIDI note number and frequency. These layers of abstraction are difficult for beginners who have never been exposed to these concepts.

Addressing Sound Design and Mapping Challenges

There are different ways of addressing these challenges. Any changes in the software environment should be considered with their tradeoffs. For example, a block based programming environment could be used, where only legal connections are allowed. However, any new tool intended to introduce students to these activities should be in the style of programming seen in popular audio programming environment like Pd, Max/MSP or ChucK.

A GUI based note selection tool that uses a piano keyboard layout or musical staff might help users program the notes they want their instrument to play. Such a tool would lower the barrier to selecting notes but misses an opportunity to engage students with this important concept in computer music.

Parameter ranges could all become normalized to use the same values so that regardless of what effect parameter a signal is connected to, it would exercise the full range. This would result in easy to map parameters but would limit opportunities for customization in sound design.

All of these ideas are feasible and would lower the barrier to working with sound design. However there is a significant tradeoff in both opportunities for customization and relation to practice. As new forms of scaffolding are introduced it is important to consider how the environment can support users in understanding the role of the scaffolding so that it can eventually be removed.

5.2.2 Approaches to Interface Design

As with the sound design and mapping, a wide range of approaches were used in these case studies for designing and creating interfaces to control sound. Diverse construction materials and electronics along with the unique contexts of each case yielded a variety of design techniques and interface styles. Here I will discuss how the Sound Clippys Platform enabled this range of results.

Making Digital Inputs

In all of these cases the introduction to interface design was the *paper piano* project. With a pair of scissors, construction paper and copper tape, students built circuits and working interfaces to use with the Sound Clippys Platform. By far the most common approaches used to create digital inputs grew directly out of this project, using copper tape to build circuits to create buttons or switches. These basic materials proved to be

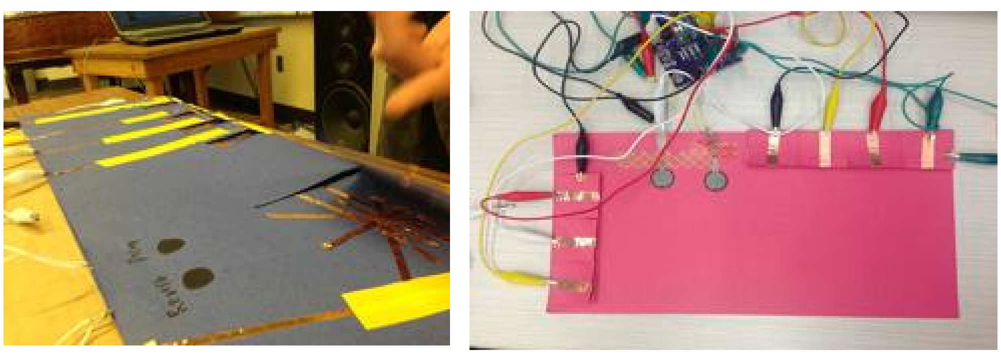

Figure 5.8 - Extensions of the basic paper piano

extremely versatile.

Some students simply refined their approach and carefully created new paper piano interfaces (Figure 5.8). Others created larger versions of the paper piano project, using cardboard instead of construction paper. Cardboard offers size and structure that construction paper does not, but it is still easily manipulated into different shapes with scissors and tape. The instrument in Figure 5.9 demonstrate the larger scale and shapes possible with this approach.

Figure 5.9 - A cardboard foot piano

Instead of copper tape, wire was used to build digital inputs for some instruments. The wire is shielded which helps avoid creating unintended connections. It is robust, allowing it to bend and be pulled without breaking. Wire allows more movement and larger scale projects as well. Examples of this include a strumming interface where strips of copper tape are strummed with a wire, a set of rings connected to a cuff, an interactive painting and a dance floor (figure 5.10).

Figure 5.10 – Shielded wire supports movement, extending reach and other switch designs: A strumming interface; finger rings; interactive painting; mini-grand piano (clockwise from top left)

A smaller number of students explored alternative conductive materials to create digital inputs. One student wondered if the metal on her tap shoes would work in a circuit (they do). Another student covered a rock with aluminum foil and suspended it inside a tube to create a switch that would trigger a sound when the tube is shaken (figure 5.11). Another is the augmented cello described above, where the strings of the cello are used in the circuit. These examples demonstrate how this simple concept of digital switches can be extended to conductive materials found in everyday objects.

Figure 5.11 – More digital input designs: Interactive dance floor and tap shoes (left); a shake sensor made with a rock covered with aluminum foil (right)

Additional Digital Inputs

The digital inputs are configured to work for passive digital switches. This design limits digital inputs to buttons and switches, but as we have seen, this can take on many different forms. Additional sensors that work in this way can easily be connected to the Sound Clippys Board. This is seen in the Musical Whether Station project, where a rain gauge and anemometer (wind speed sensor) were used. Both sensors work as switches that open and close. The rain gauge has a small reservoir that fills up as rain falls, and tilts it to the other side once full. The anemometer closes and opens a switch once per revolution. In both cases the two wires from the sensors were simply connected as a paper piano switch, with one side going to ground and the other to a digital input. Digital sensors that use SPI, I2C or other communication protocols cannot be easily used with the Sound Clippys Board. If a student is ready to use such sensors then they are likely ready to use a more traditional “breakout” approach with an Arduino and breadboard to build the circuits needed for these sensors.

Using Analog Inputs

While digital inputs have one basic circuit (a switch), analog inputs are more varied. The Sound Clippys Board has more than one style of analog input, variable resistors and 3-pin analog inputs. The 3-pin inputs work with passive analog sensors (e.g. potentiometers) and active sensors (e.g. Infrared proximity sensors).

Although analog sensors provide the nuanced control over sound needed to create expressive musical instruments, they were used less frequently than digital inputs. Across all the case studies, 18 projects used analog inputs compared to 35 that used digital inputs, out of 45 projects (these categories are not mutually exclusive). Of these, nine used variable resistors and nine used 3-pin analog inputs.

Using Variable Resistors

In all of these cases I demonstrated variable resistors by connecting bend sensors and force sensors to the Sound Clippys Board to control an audio parameter. With this basic introduction, students explore analog sensors on their own with different example patches. Different approaches are used when the students add analog sensors to their instruments. Some simply taped a bend or force sensor to their instrument. These sensors are easy to connect to the Sound Clippys Board and give students a "handle" they can map to different sound parameters.

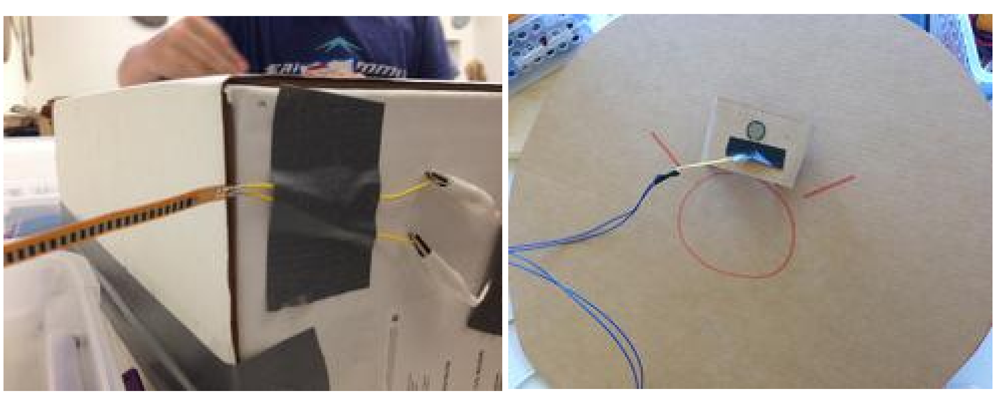

Figure 5.12 - Quick techniques for attaching analog sensors

The variable resistor input is also very good for exploring resistance in everyday materials. During the *Crafting Musical Controllers* workshops at the library I introduced a new exercise for exploring analog inputs. We created analog inputs using soft graphite pencils (available in art stores) and paper. This exercise allows students to draw sensors, and explore variable resistors with common materials. By connecting this to the pitch of a synthesizer students can listen to the change in resistance as they move alligator clips around on the paper (Figure 5.13 (left)). Another example of exploring resistance in everyday materials was in *Music Focus* where carrots were used as an input device (Figure 5.13 (right).

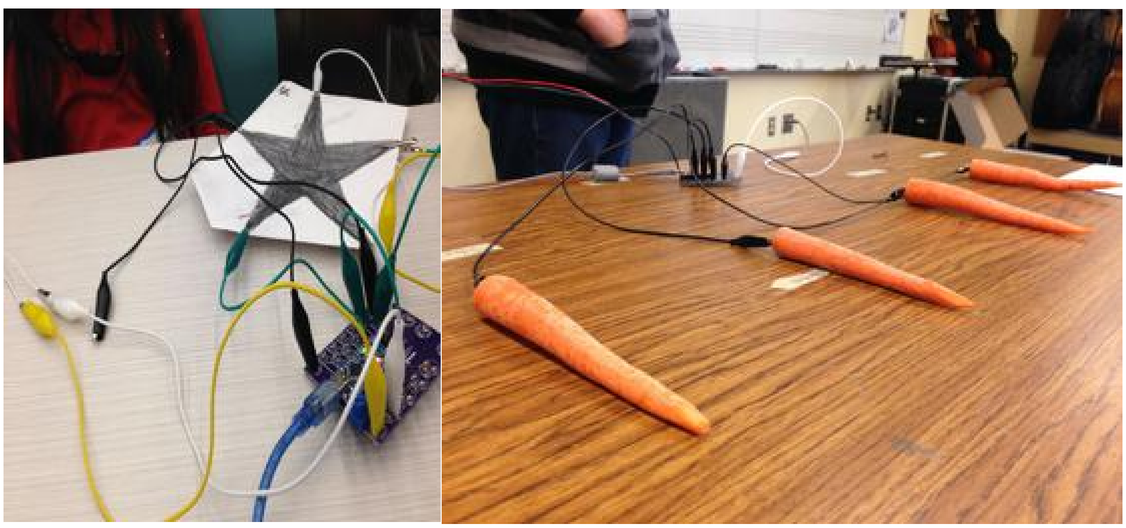

Figure 5.13 – Alternative variable resistors: made with graphite and paper (left); Carrots connected to analog inputs (right)

Variable resistors can also be integrated thoroughly as with two *Music Focus* projects that used bend sensors. One integrated the sensors into a glove, another added the sensors to a stuffed animal (Figure 5.14).

These examples show how the variable resistor inputs on the Sound Clippys Board have been used to add expressive inputs that can be easily integrated to create a range of interfaces.

Figure 5.14 – Bend Sensors in different contexts: Glove based controller with bend sensors (left); Sock monkey stuffed animal with bend sensors in the arms (right)

Using 3-Pin Analog Sensors

The 3-pin analog inputs are more closely related to the *breakout* model, as they do not incorporate an onboard circuit, but rather simply present a port that provides 5V and ground and an input that is connected directly to the analog input of the Arduino. While these inputs are arguably more flexible than the variable resistor inputs (a variable resistor could still be used here with an external circuit), in practice their use was much more homogeneous. Of the nine projects that used the 3-Pin analog inputs, seven of them used an IR proximity sensor, one used a potentiometer, and the last was for the *gas sensor sonification* project that used a series of analog gas sensors.

Analog Input Challenges

The fact that so few projects incorporated analog inputs relative to those that used digital inputs indicates challenges to using the analog inputs with the Sound Clippys Platform. One is that there are two different ways of creating analog inputs. When I created the board I considered the variable resistor inputs to be versatile and important to allow easy connection of variable resistors. The 3-pin inputs would provide the opportunity for expansion since it can be used with just about any other 5V analog sensor. However, the use of these two different input styles may confuse beginners who are being introduced to circuits for the first time.

Another challenge is knowing how to use the 3-pin analog input. The non-polarized input makes it possible to connect a sensor incorrectly, reversing the power and ground. Active sensors can potentially be damaged if plugged in the wrong way. Despite regular reminders to orient the cable so the red cable goes to 5V and the black cable goes to ground, students regularly plugged sensors in wrong, resulting in confused students and fried IR sensors. An early version of the Sound Clippys Board used a polarized connection based on the design of Blikstein’s GoGo Boards. I changed the design to the non-polarized 3-pin connection to make it easy to use small potentiometers without additional wiring. This change left open the possibility of connecting sensors incorrectly to support this readily available component.
Another challenge of using analog sensors relates to the mapping challenges discussed above. When students use an analog input they must also map it to an appropriate range and connect it in software.

5.2.3 Instrument Styles

While the range of digital and analog inputs supported by the Sound Clippys Board is limited, it has shown to be versatile in enabling a broad range of project styles. I categorized the projects produced in these case studies to understand the range of instruments produced. These results indicate two things: (1) the style of project that can be easily produced with the Sound Clippys Board, and (2) the style of project that students in these cases were drawn to making. The resulting instrument categories from these four case studies are given in Figure 5.15.

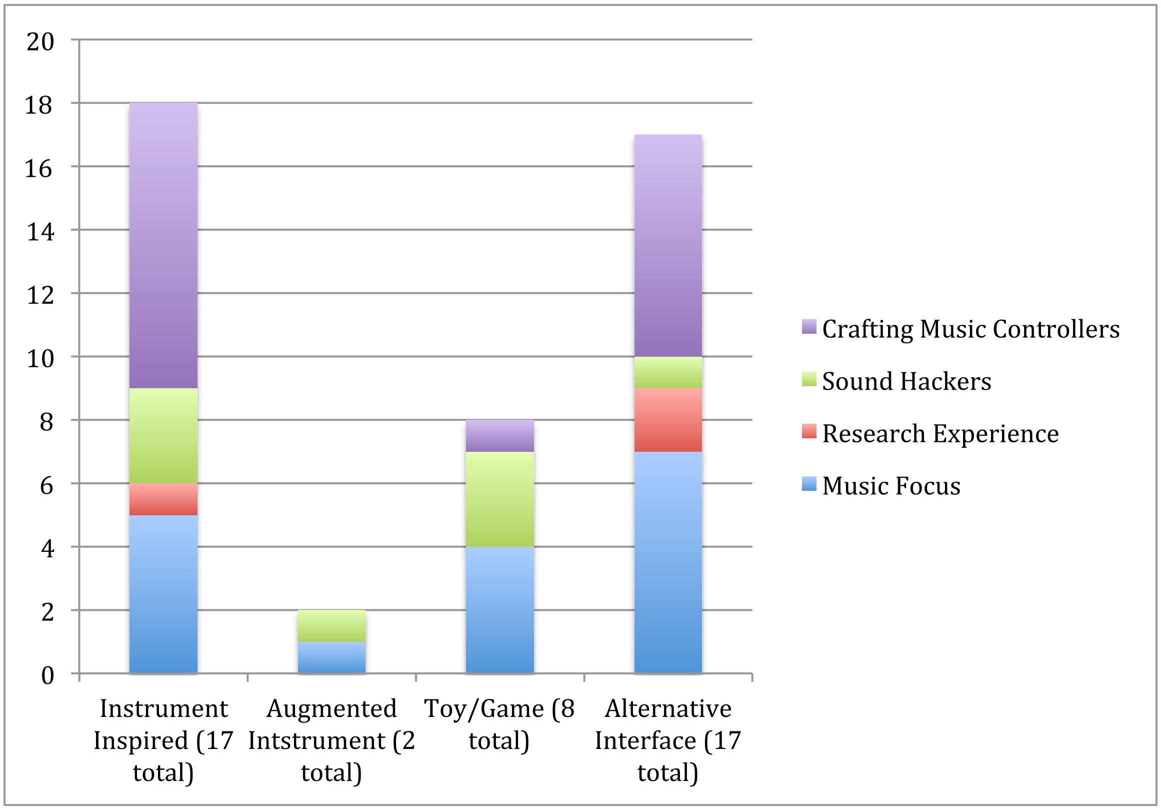

Figure 5.15 - The number of instruments in each category from the Sound Clippys case studies

The existence of these instrument categories in these results tracks with the professional practice found in the nime research community. Instrument inspired projects were the most common. These interfaces are perhaps the easiest style of instrument for beginners to relate to. The alternative interface projects are a diverse collection of interfaces that represent a range of human computer interaction devices that produce sound. These show how the Sound Clippys Platform can introduce users to new ways of interacting with computers through hands-on making. The prevalence of the toy/game category points to the potential playful nature of DMI design and reflects Cook's guiding principle that *everyday objects suggest amusing controllers*. While only two augmented instruments were created, I'm encouraged that students pursued these ideas on their own. Creating an augmented instrument requires that instruments be available, which was not always the case in these case studies. A future study that focuses on augmented instruments would illuminate the potential and difficulties in using the Sound Clippys Platform to augment existing instruments.

5.3 Conclusions

The Sound Clippys Platform was used in four case studies to explore how it can support DMI design, enabling versatile applications and compatibility with current practices within the NIME community. Despite the challenges described, students succeded in creating a wide range of DMIs.

The platform's approach to building inputs was productive in supporting the creation of diverse input devices. However, only a few students successfully developed their own patches for sound design and mapping. The students who were able to develop more sophisticated software patches and mappings spent significantly more time with Pd and the Modular-Muse library. Alternative software solutions are likely to alleviate some of the challenges encountered and enable more students to find success in developing their own software. However, in order to enable a pathway to expertise, care should be taken so that new design tools are compatible with ways of working found in artistic and research communities where DMI design takes place.

Because the platform components can be used independently, the three tasks of DMI design can be taught and explored independently. This is enabled by leveraging example designs in software, and a playable hardware board that can be used standalone without additional circuits. This enabled beginners in the *Crafting Music Controllers* workshop at the library to drop in and participate in any of the individual modules without previous experience.

Finally, the diversity of artifacts produced in these cases exemplify the *wide walls* needed for productive creativity support tools. The scaffolded support provided by the Sound Clippys Platform demonstrates compatibility with existing approaches to DMI design while enabling versatility in the results.

Chapter 6 Supporting Sound Actuation

Sound actuation and the development and use of musical robots enable new musical ideas. Sound actuation offers alternative approaches to generating sound beyond traditional techniques using loudspeakers. Automated musical instruments such as orchestrions and player pianos, have been around for centuries. Modern approaches have shifted to digital technologies, where electromechanical technologies allow digital control over acoustic sounds. These approaches use motors to produce sound by striking, shaking and otherwise producing acoustic energy in musical instruments and everyday objects.

Related works described in Chapter 2 show how sound actuation can be used by musicians and artists to enable new musical experiences. These include sound art installations, recorded music, and live musical performances. However, technical challenges of working in this space limit the development and creative exploration of new musical robots to a relatively small group of researchers and artists. Existing approaches require specialized skills to create and control these modern musical robots.

To make sound actuation more accessible I developed a musical robotics platform called *MIDI Bots* for creating new musical robots. This platform consists of a hardware board for driving motors with MIDI control, the *Modular-Muse Motor Driver Board (MMMD Board)*, and a series of mechanical designs for creating versatile sound actuators that can be controlled with existing music software. This work focuses on simplifying the development of new musical robots with the aim of broadening participation in this area. The key contribution is the MMMD hardware, which uses MIDI over USB to drive motors to make this approach to making music accessible. This platform is a step toward enabling a proliferation of musical robots. A long term goal of this work is to *make musical robots as accessible and usable as standard music production gear*, providing musicians with new means of creative expression. Democratizing musical robots could have a significant impact on how musicians compose, practice and perform music.

This chapter describes the MIDI Bots platform design, including the design goals, technical details and use. By scaffolding three components of building musical robots (electronics, software, and motor attachment) the MIDI Bots platform lowers the barrier to design and experimentation with musical robots. By supporting the MIDI protocol, the platform is compatible with existing music software environments to drive a range of solenoids and DC motors. The platform has been used for performance, installations, and in workshops described in the following chapter.

6.1 Design Impetus

The MMMD boards build on ideas from the Kitsch-Instrument to create a flexible hardware solution for musical robotics. The initial impetus for the MMMD Boards was to put on *SolidNoise,* an event I proposed that was part interactive art installation and part performance. The goal of the show was to develop a musical performance centered around sound actuation. *SolidNoise* was supported by a $1000 grant that was used to pay for the hardware development and instrument materials. Additionally, the ATLAS Center for Media Arts and Performances (CMAP) was scheduled to host an artist residency with installation and sound artist Trimpin (one of the early pioneers discussed in Chapter 2), in November 2015. *Solidnoise* was scheduled ahead of the Trimpin event to develop instruments that could be used again for the performance with Trimpin. The platform designs described here were enabled the creation of the instruments used in three performances and detailed in Chapter 7.

6.1.2 Design Goals

There are a number of challenges to creating musical robots that I wanted my hardware to address. First, Many musical robotics applications call for a large number of motors. Most of the motors used for sound actuation require high voltage and/or current power supplies. In order to create expressive musical robots with these motor drivers it is essential to be able to vary the power to the motors, enabling a range of expressive, musical sounds. Connecting these motors to physical objects is a challenge, to address this I wanted to identify some easy to create and use mechanical designs. Finally, in order to create a tool that is useful for musicians it is important that the hardware use familiar interface. The MIDI protocol is ubiquitous across nearly all music software including production tools such as digital audio workstations and music programming languages.

Given these requirements and constraints I wanted to create a hardware interface that:

1. Supports the MIDI protocol to switch motors on and off
2. Supports MIDI velocity control
3. Drives large numbers of motors with a range of power requirements
4. Can control multiple motors
5. Versatile hardware mechanisms

6.2 MIDI Bots Platform

There are several aspects to creating and controlling a musical robot. The MIDI Bots platform simplifies some of the steps while leaving room for creativity in how exactly they are applied. To achieve this the MIDI Bots platform scaffolds three aspects of building

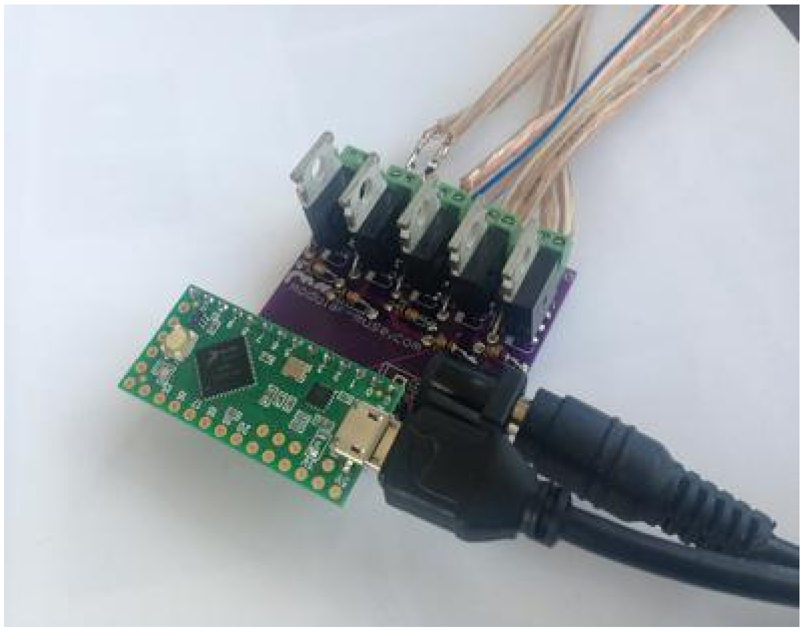

Figure 6.1 - The MMMDx5 Board with Teensy LC

musical robots: electronics, software, and motor attachment. The platform provides support for these technical aspects while leaving room to explore the acoustics of everyday objects and musical instruments.

6.3 Modular-Muse Motor Driver Boards

High power MOSFET transistors on the MMMD boards switch the power sources that drive the motors. DC Jacks supply power for the motor driver circuits. Motors connect with two wires each in screw terminals on the board.

The MMMD Boards do not use an H-Bridge circuit, which would allow controlling the direction of the motors. I decided this for two reasons: (1) most motors for sound actuation don't require bi-directional control (e.g. solenoids, DC and vibration motors) and (2) Bi-directional motors add complexity to the mechanical design and means of control, working against the goal of ease of use in order to lower the barrier to entry. Two different versions of the MMMD Boards have been created with different channel counts. The MMMDx5 (Figure 6.1) can drive up to five motors. The board is a shield for the Teensy LC, allowing it to plug directly into the board to control five motor outputs with pulse width modulated (PWM) control. This board is a compact, standalone MIDI controlled motor driver that can drive up to five motors from the same power supply with MIDI over USB. The MMMDx5 board costs around $15 for the PCB and the circuit components. The MMMDx5 shield with the Teensy LC can be assembled for less than $30, or about $6 per channel.

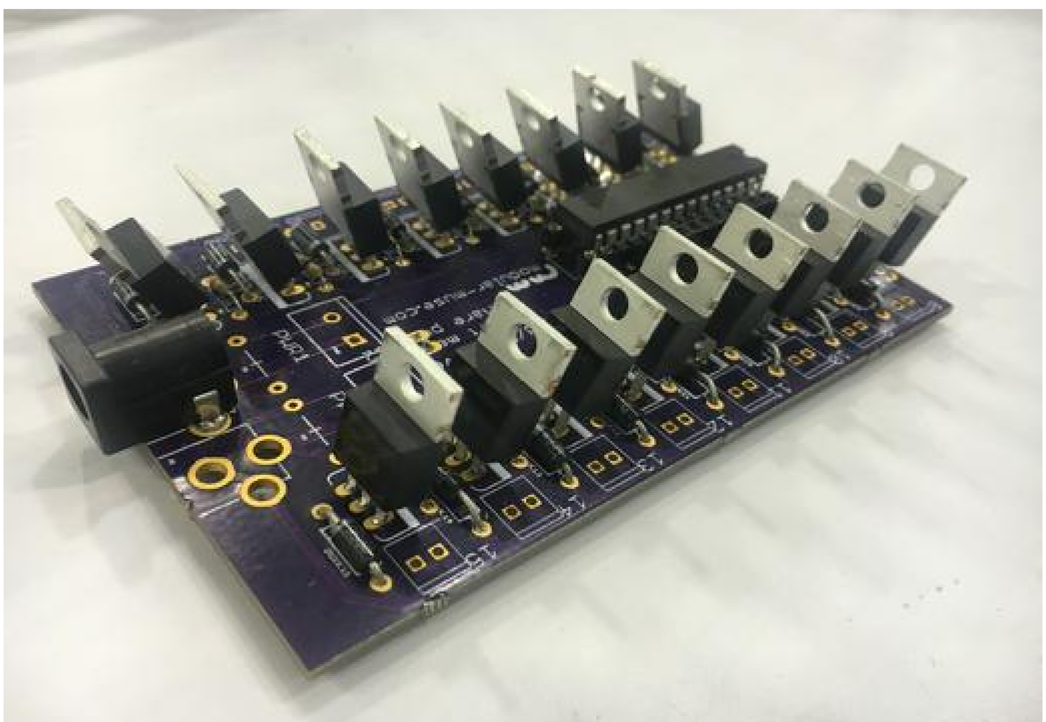

Figure 6.2 - The MMMDx16, chainable 16 channel motor driver board

The MMMDx16 (Figure 6.2) is built around the TI TLC5940, a chainable LED driver chip with 16 PWM outputs. Up to 40 TLC5940s can be chained. The cost of one MMMDx16 board is approximately $45. With a Teensy the cost for a single MMMDx16 board is around $4 per channel.

The 16 motor outputs can powered by a single power supply or with two independent power supplies, with each supply driving eight of the outputs. An optional jumper connects the two power rails to drive all 16 outputs with a single power supply. This provides flexibility in how the motors are powered. It enabling support for different voltage and current requirements for two groups of eight motors. For motors that consume a lot of power it increases capacity by supporting multiple power supplies.

6.3.1 MIDI Support

The Modular-Muse Motor Driver (MMMD) boards use MIDI over USB for control. Each board has its own program for a Teensy microcontroller that enable the boards to work without additional software (e.g. serial to MIDI converter), or hardware (e.g. external MIDI interface)[28]. The firmware relies on the PJRC MIDI library and is programmed with the Teensyduino add-on to the Arduino environment. Built in MIDI drivers allow any music software with MIDI support to trigger and control motors. This includes popular digital audio workstations like Logic Pro, Pro Tools, Ableton Live, or Presonus Studio One, as well as audio programming languages where MIDI is ubiquitous. The MMMD boards can also be connected to an Arduino or other microcontrollers .

[28] Boards and firmware: http://modular-muse.com/?page_id=525

Triggering Motors

MIDI note messages are typically used to specify musical pitch (Figure 6.3). For the MMMD Boards the MIDI note messages are used to control motors. The default program uses MIDI note number 60 to control motor channel 0, MIDI note number 61 controls motor channel 1, etc.

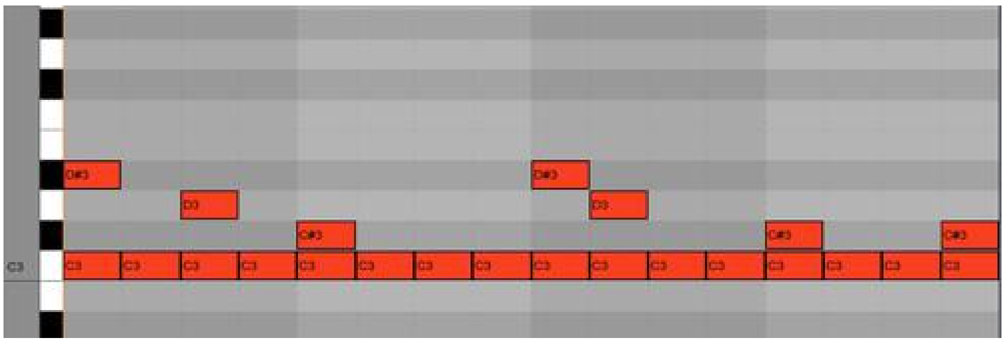

Figure 6.3 - A MIDI sequence created in Ableton Live

The MIDI note message is accompanied by a velocity message. This is a number between 0-127 and is used to control volume. The MMMD Boards translates the velocity message into a PWM output value to control the speed of a motor. The 7 bit MIDI velocity is mapped to a 12 bit PWM output. I use a simple linear mapping by bit shifting the MIDI value by 5.

Trigger and Sustain Modes

Different motors require different approaches to control. When used for percussion instruments, solenoids need to be powered long enough to move the plunger (the small metal rod that moves) but not so long that it will stay in contact with whatever it strikes. The time needed to engage a solenoid can range from 5 to 20 ms. Meanwhile, vibration motors can be used to create sustained sounds where they may be on for several seconds. In order to support driving motors in these different ways I made two versions of the Teensy firmware, one that automatically uses short triggers and another that allows sustaining power to the motor.

When using trigger mode, the microcontroller automatically turns off the motor power after a set amount of time. This "on-time" is set by a variable in software to accommodate different motor sizes or uses. The plunger in a small solenoid has a shorter distance to travel than in a large solenoid. Because of this, the smaller solenoid needs less time to fully engage so the programmed on-time should be less. This auto-off feature can save motors from accidentally being left on and burning out.

Sustain mode will leave the power to the motor at the specified velocity until it receives a note-off message. This allows length of a note to be controlled from a digital audio workstation, just as you would for any sustained note in a MIDI sequence.

6.3.2 Powering Motors

The power jack on the MMMD boards makes it easy to plug in standard DC power supplies to drive motors. More commonly, motor driver boards built for the hobby electronics market use screw terminals to attach power. This approach is convenient if you're using a bench power supply. Otherwise, to connect a standard wall wart with screw terminals requires cutting the end of a power cable, identifying the + and -, and stripping to the bare wire. The MMMD boards have an optional screw terminal port. The DC Jack has proven convenient and versatile to quickly connect a range of off-the-shelf power supplies.

6.4 Mechanical Attachments

Attaching motors to instruments and other physical objects to convert the motion of the motor into an acoustic is a challenge. The solenoid plunger can be used to directly tap objects, or the linear motion can be translated with a mechanism such as a lever. The first approach requires that the plunger be close enough to make contact with a surface when triggered. For a small solenoid this is as little as 3.5mm. The final components to the MIDI Bots platform are the mechanical designs for mounting two easy to find and inexpensive solenoids. I call these designs the Tapper and the Striker.

I have created a wide range of instruments using the MMMD boards with different motors and mechanical approaches. Through developing these automated instruments I have found the Tapper and the Striker to be very versatile. They can be easily attached and used with a wide range of instruments and objects. They are easy to produce with a laser cutter, an increasingly common tool in makerspaces and fabrication labs. The parts are readily available and inexpensive. Between Amazon and two popular DIY electronics companies, Sparkfun and AdaFruit, both the small solenoid used in the Tapper and the larger solenoid used in the Striker are available for under $5.

6.4.1 The Tapper

The Tapper design grew out of a simple PVC pipe based mount created for the Kitsch-Instrument (Chapter 3), and has evolved into an easy to assemble and easy to use sound actuator (Figure 6.4). The base of the Tapper uses two layers of acrylic. The bottom layer of 3mm acrylic acts as a spacer between the bottom of the solenoid and the acoustic surface, allowing the short throw solenoid to make contact. The solenoid rests on this layer with a second layer with a rectangular hole the solenoid fits into; this layer stabilizes the solenoid.

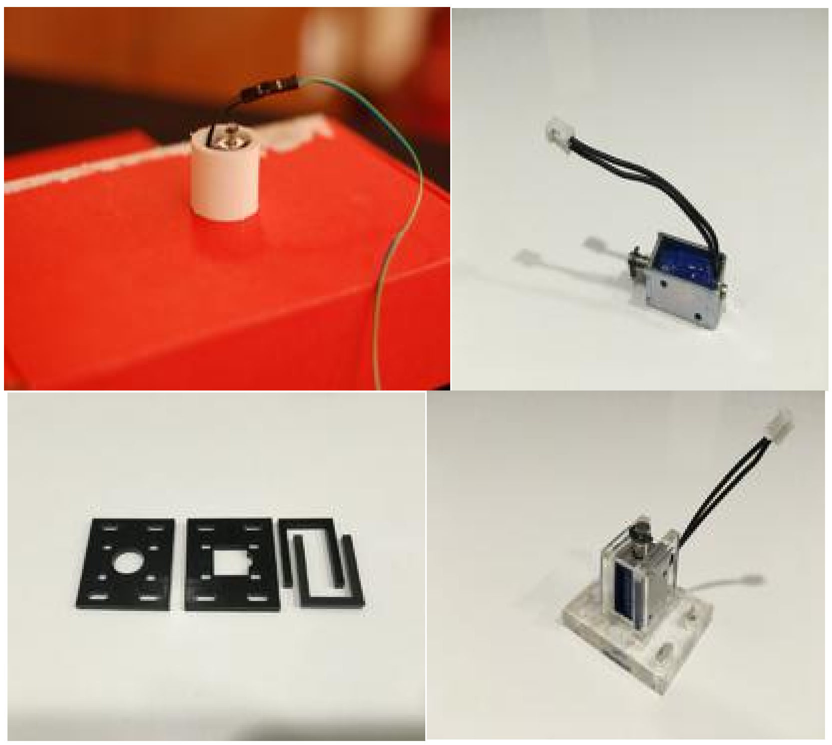

Figure 6.4 – The Tapper solenoid mount: PVC mount from the Kitsch-Instrument; small 5V solenoid used in the Tapper; laser cut Tapper mechanism parts; assembled Tapper (clockwise from top left)

The PVC design from the Kitsch-Instrument and my initial laser cut design required tapping threads into the small solenoids. Once tapped, screws were used to attach a mount to the solenoid. However, tapping the solenoids was cumbersome and expensive, as I broke the brittle taps or damaged the copper windings of the motor. My current version uses two laser cut brackets to hold the solenoid in place. This four piece design can be quickly assembled to create a robust mount (Figure 6.4).

Because the Tapper mount holds the solenoid up it can simply be placed on horizontal objects, making it easy to explore different sounds by moving it around. Holes on the edges of the base allow it to be mounted with string, zip ties, screws, etc. (Figure 6.5).

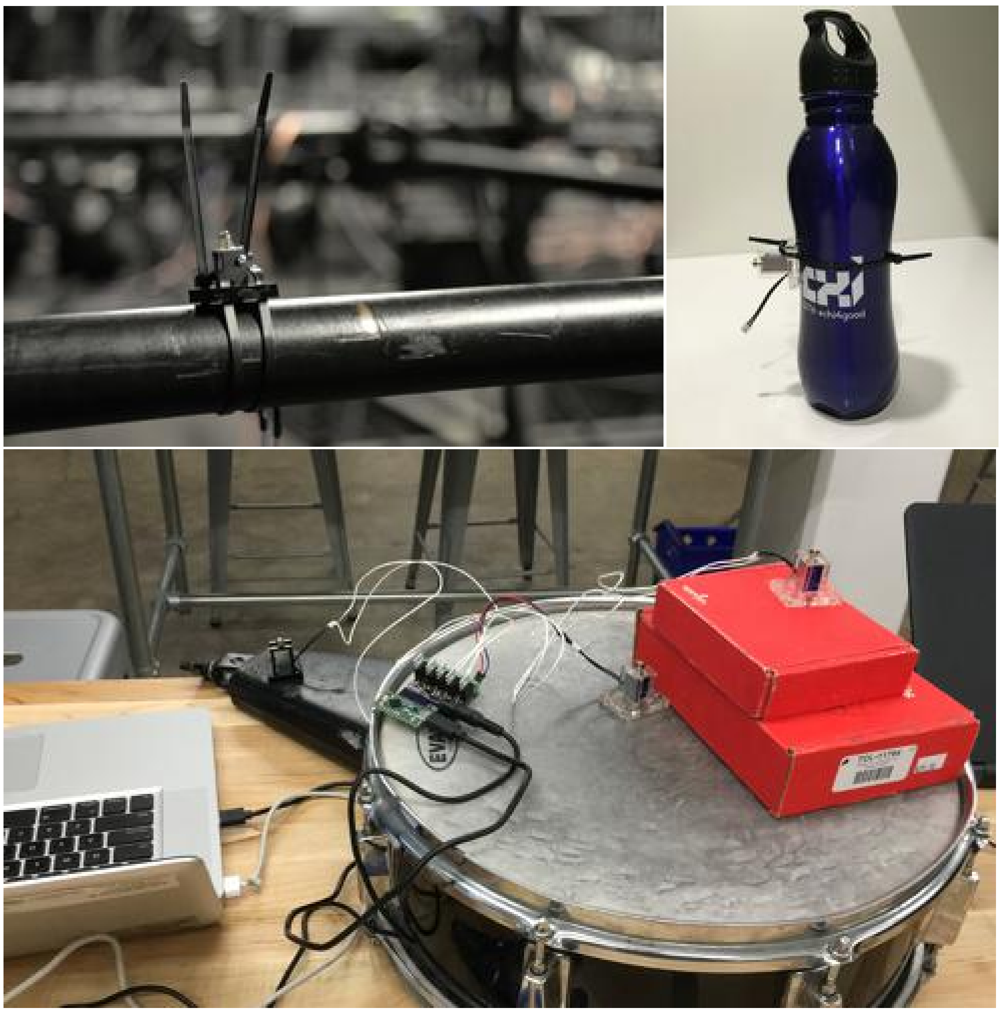

Figure 6.5 - Attaching the Tapper: with zip ties (top); Resting the Tapper on different instruments and objects (bottom)

6.4.2 The Striker

The Striker also uses a larger 12V solenoid. The solenoid is manufactured as a car part, used for electronic door locks. My design is based on a similar design found on

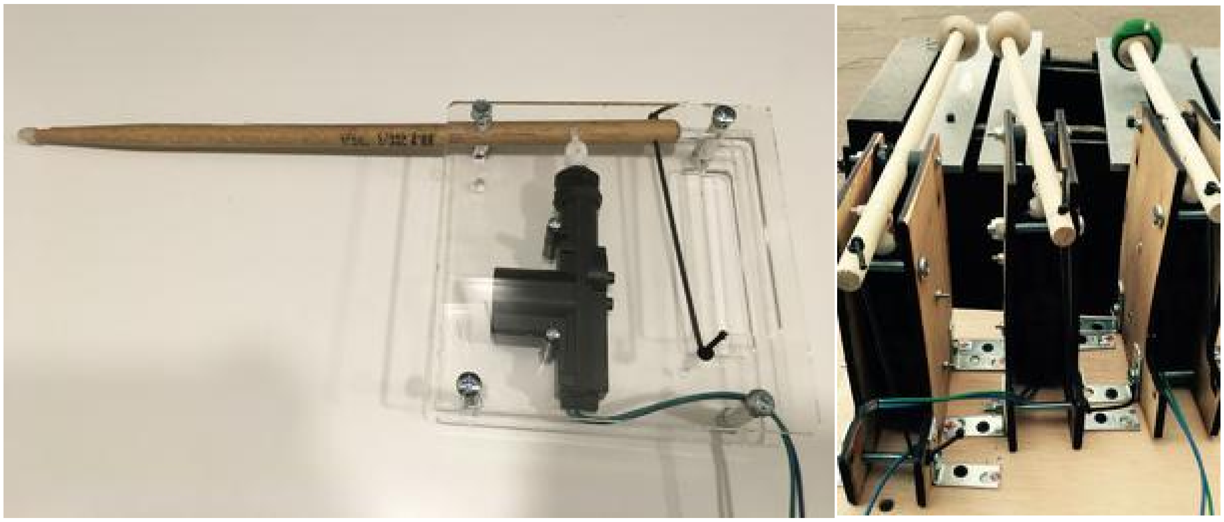

Figure 6.6 - The Striker mechanism: with handle (left); Mounted to play xylophone bars (right)

instructables.com[29]. This design uses a lever mechanism to convert the linear motion of the solenoid to a rotating drumstick (Figure 6.6). I made some changes to simplify the use of the mount used in the Instructables design. The original design has a small wing that seats the mount to the edge of a drum shell. I removed this wing to allow it to attach to a wider range of objects that don't have the contour of a drum shell. The instructables project uses a bi-directional motor driver to control the solenoid. However, this solution costs $25 per channel compared to $4-$6 per channel with my design. By adding a simple return spring (an elastic cord) the Striker design works with the uni-directional MMMD hardware.

[29] http://www.instructables.com/id/Arduino-Controlled-Robotic-Drum/

Figure 6.7 - Manually controlling the Striker

Another update to the design came while preparing for a demonstration at the CHI conference (Harriman et al, 2016). I wanted to allow visitors to my booth to move these larger mechanisms around to different objects, as with the Tapper. I added rectangular holes at the back of the mechanism to create a handle, making it easy to hold and manually move around (Figure 6.7). This approach also gives the user control over what and where the actuator strikes. This design was used at the ATLAS Spring 2016 Expo, during the CHI (2016) Interactivity sessions and during one of my workshops (Figure).

6.5 Summary

The MIDI Bots platform supports the creation of new musical robots with MIDI control of motor and versatile mechanical attachments. The MMMDx5 board is a Teensy LC shield that can control 5 Solenoids and DC motors, while the MMMDx16 can drive 16 motors and can be chained for increased capacity. The open source designs lower the cost per channel relative to current solutions and lower the barrier with MIDI support and versatile mechanical designs. The MIDI Bots platform has been used in performance, installations, and in demos and workshops in different configurations to produce a range of new automated instruments and to engage beginners in designing and controlling musical robots. These experiences are described in the following chapter.

Chapter 7 MIDI Bots Platform Results

The MIDI Bots platform has enabled the creation and use of a number of new musical robots in a range of settings. This chapter describes how the platform was used for performance, art installations, and workshops. The platform allowed designers, workshop participants, musicians and audience to participate in a range of sound actuation activities. The boards have been used with different motors and different sound actuation techniques. I show how the Tapper and Striker mechanisms make a productive platform for introducing sound actuation techniques to beginners and for quickly creating versatile instruments for performance and installations. I describe a series of instruments used in live performances, the diverse applications of MIDI Bots in a collection of interactive and new media art pieces, and two workshops. The details of how the platform components are used in these settings demonstrate new possibilities enabled.

7.1 Instruments for Live Performance

The MIDI Bots platform has been crucial in enabling efficient deployment of sound actuation techniques for several performances. Here I describe instruments created for three performances. The first performance in October 2015 was the *SolidNoise* event. A month later the platform was used again to create two new instruments for a performance with Trimpin. The platform was also used in February 2016 performance at the Georgia Tech Guthman New Musical Instrument Competition.

7.1.1 SolidNoise

The production of the SolidNoise ensemble began in August 2015. The event was put on by a team of seven musicians, engineers and designers. SolidNoise made exclusive use of sound actuation of everyday objects (as opposed to existing instruments) to produce sounds. The ensemble included percussion, wind, and string musical robots. The design and fabrication of instruments and musical compositions were created over a relatively short period of eight weeks.

Early design meetings focused on exploring the acoustics of everyday objects, including sheet metal, trash bins, and plywood. We used the MMMD Boards to drive DC motors and various solenoids to tap, strike, and scrape objects. Low-tech prototyping techniques including duct-taping motors, and holding and moving actuators manually supported the hands-on explorations. As we tried different materials and instrument ideas to explore musical possibilities. The result was a complementary ensemble of percussive and sustained instruments with a variety of sound textures.

For the performance, we used six different instruments. Four of the instruments used the MMMD boards, while two used standard audio amplifiers to drive electromagnets and tactile transducers to generate sound. The robots that utilized the MMMD boards included the Tappers, and Strikers described in Chapter 6 and two more instruments referred to here as Shakers and Blowers (Figure 7.1).

The Shakers use vibration motors attached to sheet metal, shaking it to produce a rumbling thunder like sound. The Blowers use solenoid controlled air valves to blow air

Figure 7.1 - Instruments at the SolidNoise event: Strikers, Shakers, and Blowers (clockwise from top)

over the mouth of beer bottles. Sixteen Tappers were connected to the metal grid above the theater, spread across over 40 feet to create a spatial effect. The Strikers were used to create a four piece drum set including drums made with buckets, plywood and a tin can (Figure 7.1).

The instruments were spread throughout the theater, filling the space with digitally controlled acoustic sounds. We used the programming environment Max/MSP to network a group of computers using wireless OSC messaging. The instruments were controlled from a master computer as in typical laptop orchestra performances (Fiebrink et al, 2007). The slave computers received commands and sent out MIDI messages to trigger and control motors. The setup allowed control over sounds originating all over the theater. During the first hour of the event, visitors could play with the instruments using different interfaces. Following the hands-on tour of the instruments, three compositions were performed. The three pieces used different approaches to control. The first piece was entirely computer controlled, while the second relied on performers to control sounds using custom interfaces, and the final piece used a combination of human performance and algorithmic control.

7.1.2 Trimpin

The ATLAS Center for Media Arts and Performance hosted sound artist Trimpin for a performance on November 14th. Trimpin worked with a group of students, faculty, and staff in preparing for the concert, creating new instruments and composing a new piece

Figure 7.2 - Trimpin's 80/20 mounting system: shown on a piano harp installed for the 2015 ATLAS performance

to present at the performance. Trimpin brought with him his own custom motor driver hardware and mechanisms for sound actuation. He has developed and refined modular attachment system using the 80/20 extruded aluminum framing system that was mounted to a piano harp (Figure 7.2). The performance featured a range of sound actuators, including DC motors with various attachments for scraping and brushing the strings, solenoids for striking the strings, and an electromagnetic actuator.
In addition to Trimpin's instruments, 16 Tappers were again installed in the grid of the Black Box, and two new instruments were created using the MMMD Boards. One was wind chimes, triggered by pulling the chimes toward the clapper using a large solenoid (Figure 7.3). Another used two vibration motors suspended above an autoharp. The Tappers in the grid and the other new instruments created with the MMMD Boards were integrated into a performance that was composed and improvised by the group.

7.1.3 Guthman Musical Instrument Competition

A third live performance with the MIDI Bots platform took place at the Guthman New Musical Instrument Competition in February 2016. The annual competition held at Georgia Tech is a showcase for new musical inventions and performance. The *SolidNoise Ensemble* was one of 20 semifinalists out of over 100 submissions. For the competition performance fellow PhD student Abhishek Narula and I installed and performed a structured improvisational piece at the DramaTech Theater (Figure 7.4). Traveling by airplane made it difficult to bring large instruments, but the flexibility of the Tappers and Strikers allowed us to improvise instruments from objects we found at the theater and around town before the performance.

As with the original SolidNoise performance, we attached Tappers to the grid of the theater, spreading them as possible. Strikers were attached to microphone stands and adjusted relative to different drums borrowed from the Georgia Tech School of Music. Other Strikers and Tappers were attached to boxes, bottles and other found objects. The rest of the ensemble for this performance included the Shakers and a new vibration motor design being developed for Solarophone (described below).

Figure 7.3 - Performing at the 2016 Guthman Musical Instrument Competition

For this performance I used a control interfaces that used a MIDI controller, light sensors (using the Sound Clippys Board) and a GUI interface in Max/MSP to control the 26 motors. The control interface allowed me to create sequences, interact with algorithms, control tempo, and start and stop algorithms and sequences for the different instrument groups.

These performances demonstrate some important features of MIDI Bots platform including: the ability to use the mechanisms with found objects alongside traditional instruments makes the instrument groups fluid; sound design and spatial design take place in the physical domain, and compatibility with existing digital music software allows existing knowledge and expertise to be applied to composing and performing with musical robots.

7.2 Art Installations

In addition to the performances these designs have been used in a handful of art installations. Danny Rankin used it to control high powered LED lights (instead of motors) in an interactive light fixture he built for the ATLAS BTU Lab (Figure 7.5 (right). Matt Bethancourt presented *Box/Woods*, a sound art installation at the Society for Electro-Acoustic Music in the U.S. (SEAMUS) annual conference (Bethancourt, 2016). *Box/Woods* consists of 20 boxes that generate ambient sound texture based on a data set. The data is sonified with Arduinos driving small speakers and Tappers with MMMD boards (Figure 7.5 (left)).

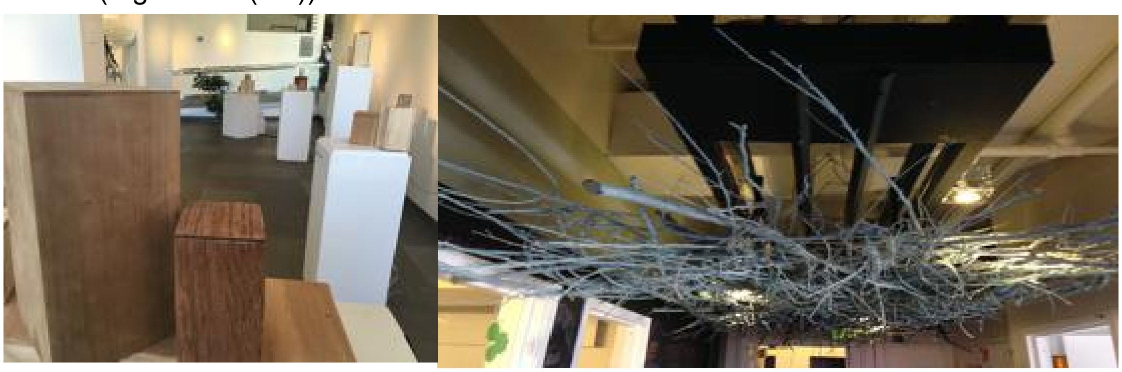

Figure 7.4 – Alternate uses of the MMMD boards: Bethancourts Box/Woods (left); Rankin's interactive light fixture (right)

For the ATLAS Expo in Fall 2015, I turned a stairway into a musical instrument that spanned three floors. I used the Sound Clippys Board to create a user interface with four light sensors. (Figure 7.6). The light sensors were used to trigger Tappers attached to the railing and wall. The sensors were mapped to control motors above when shadowing the left side, while the right side controlled the Tappers below.

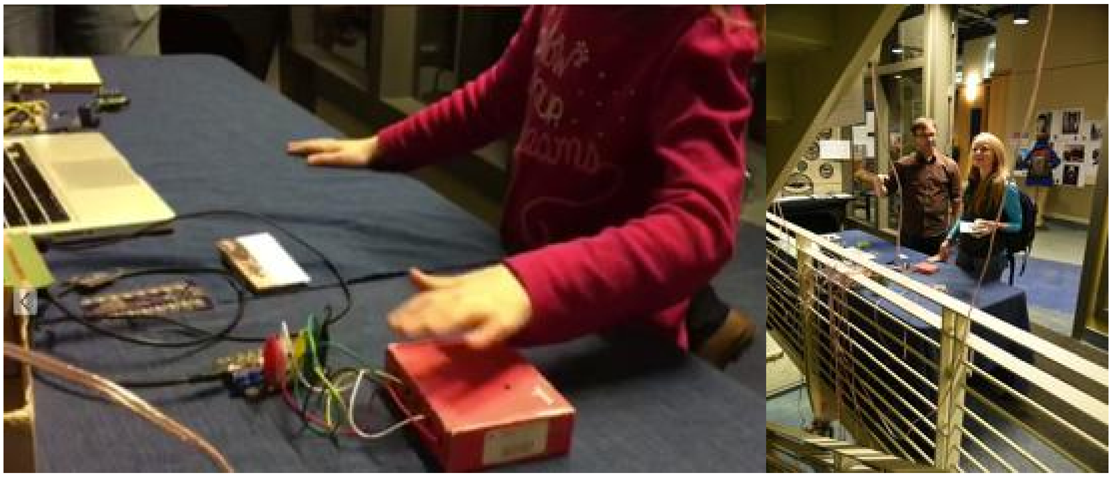

Figure 7.5 - ATLAS Expo stairway installation: Light sensor connected to Sound Clippys Board for control (left); Motors connected to the MMMD board going up and down the stairway (right)

The Sound Clippys and MMMD hardware designs enabled me to design, build and install this interactive musical instrument in a day's time. It also allowed me to prototype an installation I had proposed at the Boulder Public Library, and test it with an audience

7.2.1 Solarophone

In July 2015, around the time I received the initial MMMD Boards, I became the first *Hacker in Residence* at the Boulder Public Library. I proposed installing a sound art installation in the lobby of the building and holding a workshop on musical robotics. I was awarded a grant of $2000 for the installation to buy motors, musical instruments and MMMD boards. *Solaropohone* was installed in March 2016 and remained in the library through June 2016.

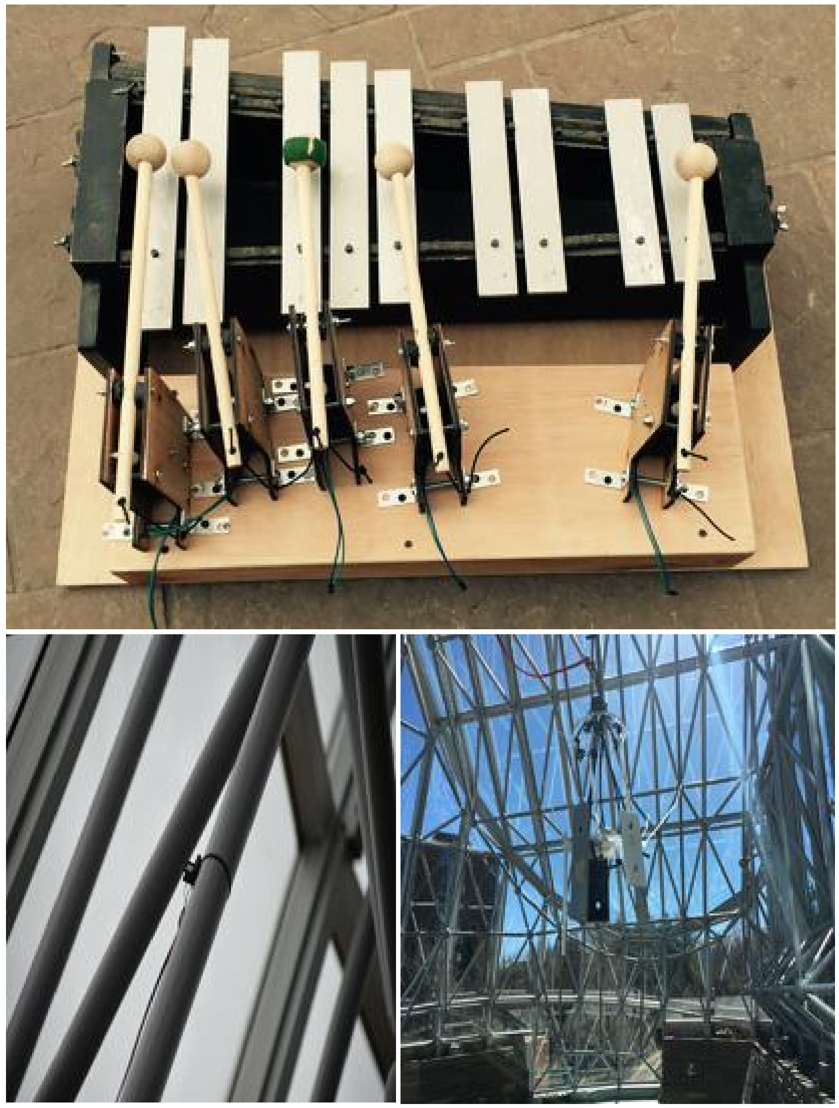

Figure 7.6 - Solarophone instruments: Strikers, Chimes, and Tappers (clockwise from top)

Like the installation during the ATLAS Expo, *Solarophone* is played by casting shadows on light sensors. The sensors trigger 37 sound actuators installed throughout the space. The instruments included 16 Tappers attached to the steel structure of the building, 16 Strikers played on four different xylophones placed around the perimeter of the space, and five Chimes (a new design) hung from the high ceilings.

The Chimes are made to excite tuned metal bars (taken from old glocknspiels) with vibration motors attached to a round acrylic disc (Figure 7.7). Each chime has five bars that are hung so they rest lightly on the vibrating disc. The notes of the tone bars are selected to create musical chords from the five notes.

The space has a glass ceiling that reaches over 40 feet high. The arched glass ceiling works as an amphitheater for the installation, reflecting the sound back down. The instruments were placed as widely as possible both horizontally and vertically to allow for spatial effects.

The amount of light in the glass entry varies throughout the day as the sun passes. First thing in the morning it shines directly on the Solarophone interface while the beams cast moving shadows. The interface eventually goes into the shadow of the building as the sun goes down. The changing light conditions affect the responsiveness of *Solarophone*.

Solarophone Interface

The interface maps a light sensor to each of the motors installed in the space. Four additional light sensors in the corners of the interface are used to create a moving average, approximating the ambient light and allowing the interface to adjust to the changing conditions.

Solarophone cycles through three different modes. The first is a simple manual mode that triggers a corresponding motor when there is a change from light to dark. The other two modes have an underlying rhythm and algorithms to produce changing musical patterns. The light sensor at the center of the interface controls the tempo. When no one has played *Solarophone* for 60 seconds it begins a sparse algorithm that plays on its own.

Figure 7.7 - Solarophone user interface with 42 light sensors

The installation demonstrated robustness for the hardware and mechanisms, running for over 12 hours a day for four months straight. The motors likely triggered hundreds of thousands of times. Over that span the MMMD hardware continued to operate as designed, as did the installed Tappers and Strikers. Unfortunately not all of the Chimes lasted for the duration of the installation. One vibration motor burned out and some of the nylon strings holding the heavy bars broke.

Interactions

I watched visitors interacting with Solarophone on many occasions during the four month installation. It never took long before someone stopped at the interface to play. The occasional sounds from above caused people to look up to find the sound source.

The size and placement of the interface allowed for several people to play simultaneously. This was often the case, especially with children who would crowd around it to get a turn.

Over the four months it was installed many regular library customers became quite familiar with it. I often saw visitors bringing friends or family over to the interface to show them how to play it. One of the library staff shared a story from a visitor who brought her grandkids in to play the installation so they could know what it's like being a conductor.

I talked to one visitor who complimented the quality of the speakers and sounds. When I told him there were speakers but that the sounds were acoustic he responded, “that's why it sounds so real”. This interaction highlights the ability of sound actuation to stimulate sensitive human hearing.

7.3 Workshops

In 2016 I held two Musical Robotics workshops using the MIDI Bots platform. During the two hour workshops I gave an overview of sound actuation and demonstrated the platform. The rest of the time was devoted to hands-on activities where participants used Pd or a digital audio workstation to control motors while exploring different objects to create sounds with.

The first workshop was held in the ATLAS BTU Lab with 17 adults. Several stations were set up with two Tappers each, and one with DC and Vibration motors. The participants were engaged in the activities and we discussed ideas about how to use the hardware and motors. The participants, mostly CU students, came from a range of educational backgrounds, including engineering, film, arts, and music. The platform enabled all the attendees to control and experiment with sound actuation.

Following the workshop I invited people to stay for an informal jam session with the musical robots. One of those who stayed was Hans Heidmann, an engineering student and musician. He shared a song he was working on and we were able to quickly change

Figure 7.8 - ATLAS Musical Robotics workshop: Jamming with robots; Tappers on different boxes; co-interacting with a violin (clockwise from top)

the drum tracks from triggering digital sounds to driving an MMMD board. Within a few minutes his song was playing back with sound actuation being used for the drum part.

The second workshop was held at the Boulder Public Library makerspace, BLDG 61. This workshop had 10 attendees ages 12 and up. Five stations were set up with different actuators. The participants used Pd and the digital audio workstation Presonus Studio One to control the motors. Two students working together created a short composition using Studio One for the first time.

In these workshops and the informal jam session, the MIDI Bots platform enabled beginners to control and explore different actuators to create sound with musical instruments and other available objects. These workshops also highlight the opportunity to explore acoustics by manipulating different objects. For instance, students would move Tappers around on a surface while they played, or would manipulate the resonance of an overturned cup by tipping it up at different angles. Compatibility with MIDI makes it easy to use the platform with existing tools. This allows students with experience to leverage tools they are already familiar with to control the motors.

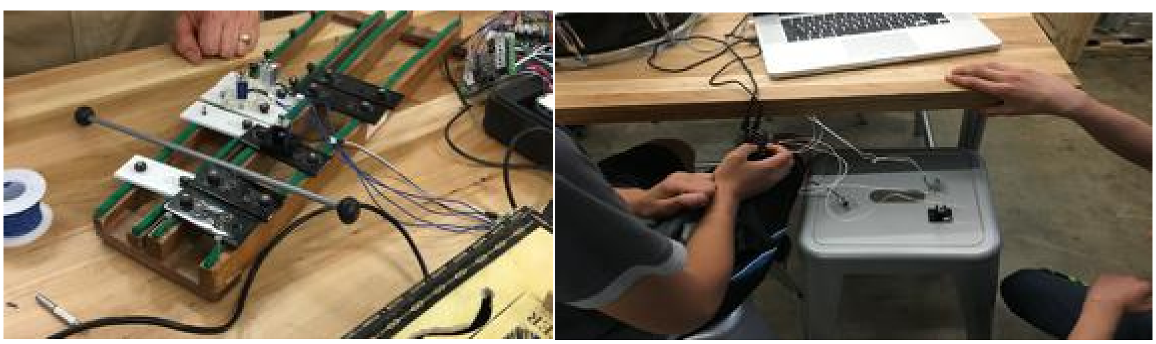

Figure 7.9 – Explorations from the BLDG 61 Musical Robotics workshop

7.4 Discussion

The boards can drive large numbers of DC and vibration motors as well as solenoids for sound actuation. The wide range of motors and attachment mechanisms presented here can be used with traditional instruments and everyday objects.

The most significant results that highlight the strengths of this approach are:

1. The mechanisms can be used with found objects alongside traditional - This makes the instrument groups fluid and unique. They can be changed based on the setting and the type of sounds desired.
2. Sound design and spatial design take place in the physical domain - This is an interesting and creative aspects of sound actuation and the MIDI Bots platform. Sounds are explored by moving mechanisms to different objects. And, rather than panning sounds in virtual space, sound come from a physical object that can be moved relative to a listener. Sound actuated instruments can also be co-manipulated as acoustic objects to shape and change the sound.
3. Compatibility with existing digital music software - This allows existing knowledge and expertise to be applied to composing and performing with musical robots. Digital musicians can apply their skills and beginners can learn how to control sound actuation with popular tools.

7.4.1 Challenges and Limitations

There are some notable limitations found with the platform. First, the usable dynamic range is limited. The simple linear mapping of the velocity from 7 bits to the 12 bits of available PWM resolution allows only a small range of corresponding motor power. In practice the motors exhibit the greatest variation in speed at the low end of the MIDI velocity values (from 0-10). While velocities from 11-127 do not produce a discernable change in motor speed. A mapping that takes this variation into account could provide a more usable dynamic range. A calibration step as described by Murphy et al (2012) would help maximize velocity dynamics.

Another limitation of the current MMMD Boards is in supporting both the trigger mode and sustain modes described in Chapter 6. Currently the boards have to be reprogrammed in order to change between these modes. Similarly, in order to update the on-time of a solenoid the boards must be reprogrammed. Some additional onboard hardware such as switches for the actuation mode and a knob for the on time would make these features more convenient and accessible.

The MMMD boards, with pre-programmed Teensy microcontrollers, can be productive for musicians without technical backgrounds, but without pre-fabricated mechanisms users are still left to solve many mechanical problems to make use of motors. The instruments produced for SolidNoise relied heavily on a laser cutter to fabricate mechanisms and mounts for our motors. Although such fabrication tools have proliferated, I suspect this would be too great a barrier for most musicians who might benefit from musical robots.

Finally, the style of sound actuation is limited by the circuits used in the MMMD Boards. The instruments presented are mostly percussive, although examples of wind and string actuation are also described using the platform. The MIDI Bots platform is well suited to driving DC motors and solenoids to produce sounds by striking, tapping and shaking objects. Other forms of sound actuation using different mechanisms and electronics suggest areas for future design tools for sound actuation.

7.5 Opportunities and Future Work

The development and use of musical robots has been predominately by a small group of researchers and artists. Tools such as the MMMD boards, and accompanying mechanisms can enable new applications for musical robots with the potential to expand the creative possibilities for a broad range of composers, artists and musicians.

Beyond the applications described here, musical robots hold potential to create new ways of making music for a wide range of users. Like a metronome they can be practice partners. They might provide a rhythm section for a band that needs a drummer. They could be used to accompany a solo artist looking for a unique backing track. Additionally, as we have seen with the Kitsch-Instrument, they can provide an engaging entrée into the world of robotics with a unique appeal to the musically inclined.

Through performances, installations, demonstrations, and workshops I have shown how the platform enables beginners to make use of sound actuation. Visitors to performances and art installations as well as workshop participants express significant interest in the new ideas enabled by these tools.

7.6 Summary

The MIDI Bots platform played a large role in the success of performances, installations, and workshops described here. In performance it enabled rapid and diverse experimentation with materials and instrument ideas. Compatibility with existing music software systems enable musicians and beginners to leverage existing skills and tools. The use of the platform in art installations show how the platform can support creative use of motors enabling new possibilities for artists and audience. The support provided by the platform allows users to focus on musical composition and the creative sonic possibilities that the actuators enable.

Chapter 8 Conclusions

8.1 Summary and Contributions

Advances in computation have expanded creative possibilities and human capabilities in the realm of music. Computers enable new sounds, and approaches to musical composition, while interactive and robotics technologies have enabled new musical instruments and sound actuation techniques under digital control. Approaches to digital sound synthesis can simulate traditional musical instruments and produce new sounds only possible with computation. In a 2008 interview computer music pioneer Max Mathews reiterated the power of digital signal processing for audio stating, “the sampling theorem shows that there are really no limits to the sounds you can make from samples. Any sound the human can hear, you can make with the right number, accuracy, and combination of samples, so *the computer is a universal instrument*” (Park, 2008).

To make use of this *universal instrument* we need new tools that enable musicians to access the full capabilities of computers. Of course, there are wonderfully powerful software tools for music, from popular consumer level software tools to audio programming environments. But the rigidity of widely used software tools, including digital audio workstations and software instruments, limit the role computers play in music. Meanwhile, open ended audio programming languages allow diverse, new and yet to be imagined uses for computers in music. And indeed, a small, but growing group of artists, technologists, and researchers are exploring the boundaries of how else a computer can be used to enable new experiences and to explore exciting musical possibilities. However, there is a significant ease-of-use gap between music programs and music programming environments. With new tools that bridge this gap we can invite musicians of all ages to participate in redesigning how computers are used for music.

My contributions include hardware and software designs with novel and useful solutions to lower the barrier to using interactive and robotic technologies for music. In particular:

1. The Sound Clippys platform enables beginners to design and build digital musical instruments. The alligator clip based Sound Clippys board allows easy use of familiar craft materials and techniques to create diverse interfaces. The Modular-Muse library support mapping those interfaces to simplified sound design tools within the open-ended Pd programming environment.

2. The MIDI Bots platform proved versatile and capable to support beginner and experienced users control sound actuation in musical performances, art installations, and workshops. Compatibility with standard USB MIDI drivers enables users to focus on musical applications of motors. The mechanisms presented can used to create sound with a range of objects including traditional musical instruments and everyday objects.

3. My design criteria highlight my approach to lower the barrier to designing for interactive music. The criteria of *support* allows new people to participate. The criteria of *compatibility* allows knowledge to transfer both in and out of these systems, and the criteria of *versatility* promotes variety by remaining open to a wide range of uses.

This exploratory work shows great potential for promoting invention and musical creativity by creating new tools to support the use of new technologies and novel techniques.

8.2 Future Work

The designs presented in this dissertation cover a lot of ground. The pursuit of a hardware and software toolkit to enable beginners to explore wide ranging applications of interactive and robotic technologies for music is ambitious. My designs have shown to be effective at enabling particular styles of design activity, but it is by no means universal. The challenges and limitations discussed in Chapter 5 and 7 suggest design changes and future studies.

8.2.1 Improvements

Sound Clippys

With my Sound Clippys platform the paper piano project provided a productive leaping off point for creatively designing with digital inputs. The simplicity and versatility of the craft materials promoted diverse creations. However, the use of analog inputs proved more challenging. The development of a pedagogical equivalent of the paper piano projects is needed for analog inputs. Additionally, for the target audience in my Sound Clippys case studies, the use of two different analog input styles was confusing. Simplifying the board to only use the *variable resistor* input will allow students to experience the concept of analog inputs as music controls without significantly sacrificing versatility.

MIDI support for the Sound Clippys board would create opportunities for sound design with a wider range of tools including off the shelf software instruments. Such an approach may be more appropriate for beginners, especially when limited time prohibits deep engagement with the software library.

MIDI Bots

The MMMD boards are particularly effective at driving solenoids and DC motors, and the Tappers and Striker mechanisms can be attached to a range of objects to produce sound by physical hitting them. But these approaches represent only a fraction of sound actuation techniques. New designs that easily enable controlling air, electromagnets and other approaches will broaden the appeal and potential design space of a sound actuation platform. A new hardware design with an interface to control parameters like the on-time and trigger mode will allow controlling a range of motors without reprogramming the microcontroller.

8.2.2 Future Studies

To create a more nuanced understanding of the effectiveness of my designs, future studies that investigate specific conditions and design approaches will help fill out the details of how to effectively support 1) specific tasks involved in these activities, 2) different audiences, and 3) particular styles of interaction design. For instance, a study focused on creating augmented instruments with experienced musicians would illuminate some particular affordances and challenges that would begin to create a more complete picture of my designs.

8.2.3 Extensibility

Both platforms are built with and within an existing ecosystem of tools. Because of this, experienced designers can easily extend their capabilities by reprogramming the microcontrollers, or identifying new hardware and software applications.

For instance, I have created a Max/MSP patch to parse the serial data stream to receive inputs from the Sound Clippys board. This opens up the hardware to an alterative and well established community of artists, engineers and musicians that use Max/MSP. I hope to leverage the extensibility of these designs to continue to expand the audience and uses of these tools.

8.3 Conclusions

By applying lessons from scaffolded educational tools and constructionist design activities to the field of computer music, this dissertation has shown how we can enable new creative applications of interactive and robotic technologies. In inviting a range of users to explore new creative possibilities, this work has emphasized the importance of the three design criteria of *support*, *compatibility* and *versatility*.

We are living at an exciting time where interactive and robotic technologies are facilitating ever more ubiquitous computing. For music and art, activities that are uniquely human, these technologies have demonstrated great potential to revolutionize the way we experience and create. However, to realize the full potential of these powerful technologies, we need to integrate it more fully into these disciplines. Music and arts training that includes technology education can inspire new forms of creative expression. The pursuit of new tools is the pursuit of new opportunities to invite and inspire a broad audience to participate in this next wave of creative expression.

8.4 Publications

Citations of publications related this work are as follows:

Harriman, Jiffer, Bethancourt, Matt, Narula, Abhishek., Theodore, Michael, and Gross, Mark. "SolidNoise: Making Musical Robots." *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2016.

Oh, Hyunjoo, Jiffer Harriman, and Abhishek Narula. "Crafting Mechatronic Percussion with Everyday Materials." *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2016.

Harriman, J. "Feedback Lap Steel : Exploring Tactile Transducers as String Actuators". Proceedings of the 15th International Conference on New Interfaces for Musical Expression. NIME, 2015.

Harriman, J. "Pd Poems and Teaching Tools". Proceedings of the 15th International Conference on New Interfaces for Musical Expression. NIME, 2015.

Harriman, J. "Start 'em Young: Digital Music Instruments for Education". Proceedings of the 15th International Conference on New Interfaces for Musical Expression. 2015.

Harriman, Jiffer, Michael Theodore, and Mark Gross. "The Kitsch-Instrument: Hackable Robotic Music." Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction. ACM, 2015.

Harriman,J. "Modular-Mus: towards an interaction design toolkit for music." Proceedings of the 2014 companion publication on Designing interactive systems. ACM, 2014.

Harriman, J., Theodore, M., Correll, N., and Ewen, H. "Endo/Exo : Making Art and Music with Distributed Computing". Proceedings of the 14th International Conference on New Interfaces for Musical Expression. 2014.

Harriman, J. "Demo hour." Interactions 21(6): 8-11, 2014.

Harriman, Jiffer. 2013. Sound actuation and interaction. In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13). ACM, New York, NY, USA, 351-352.

Harriman, Jiffer. "Sinkapater – An Untethered Beat Sequencer." Proceedings of the 12th International Conference on New Interfaces for Musical Expression. NIME, 2012.

Harriman, Jiffer, Locky Casey, Linden Melvin, and Michael Repper. "Quadrofeelia–A New Instrument for Sliding into Notes." Proceedings of the 11th International Conference on New Interfaces for Musical Expression. NIME, 2011.

Chapter 9 Bibliography

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of Learning Group*, 1–11.

Allen, Jamie. "Could this Be What It looks like? Lifelike Art and Art-and-Technology Practice." Artnodes 11 (2011).

Berdahl, Edgar, and Chafe, Chris. "Autonomous New Media Artefacts (AutoNMA)." Proceedings of the 2011 conference on New interfaces for musical expression. (2011)

Berdahl, E., & Ju, W. (2011). Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform, 2–7. Proceedings of the 2011 conference on New Interfaces for Musical Expression. (2011)

Bethancourt, M. (2016). Box/Woods. SEAMUS Annual Conference, Statesboro, GA, Averitt Center for the Arts.

Blikstein, Paulo, and Arnan Sipitakiat. "QWERTY and the art of designing microcontrollers for children." *Proceedings of the 10th International Conference on Interaction Design and Children*. ACM, 2011.

Bloland, Per. "The electromagnetically-prepared piano and its compositional implications." *Proc. ICMC*. 2007.

Bukvic, I., Baum, L, Layman, B., & Woodard, K. (2012). "Granular Learning Objects for Instrument Design and Collaborative Performance in K-12 Education", 12–14.

Cook, P. "Principles for designing computer music controllers". Proceedings of the First International Conference on New Interfaces for Musical Expression. 2001.

Cook, Perry R. "Re-Designing Principles for Computer Music Controllers: a Case Study of SqueezeVox Maggie." Proceedings of the 9th International Conference on New Interfaces for Musical Expression. 2009.

Dobrian, Christopher, and Daniel Koppelman. "The'E'in NIME: musical expression with new computer interfaces." Proceedings of the 2006 conference on New interfaces for musical expression. IRCAM—Centre Pompidou, 2006.

Fiebrink, Rebecca, Ge Wang, and Perry R. Cook. "Don't forget the laptop: using native input capabilities for expressive musical control." Proceedings of the 7th international conference on New Interfaces for Musical Expression. ACM, 2007.

Fiebrink, Rebecca, Dan Trueman, and Perry R. Cook. "A Meta-Instrument for Interactive, On-the-Fly Machine Learning." *Proceedings of the 9th International Conference on New Interfaces for Musical Expression.* 2009.

Focke, Anne. *Trimpin: Contraptions for Art and Sound*. University of Washington Press, 2011.

Greenberg, Saul, and Chester Fitchett. "Phidgets: easy development of physical interfaces through physical widgets." *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 2001.

Harriman, Jiffer. "Feedback Lap Steel: Exploring Tactile Transducers as String Actuators." Proceedings of the 15th International Conference on New Interfaces for Musical Expression. 2015.

Harriman, Jiffer, Michael Theodore, and Mark Gross. "The Kitsch-Instrument: Hackable Robotic Music." *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2015.

Harriman, Jiffer. "Modular-Mus: towards an interaction design toolkit for music." *Proceedings of the 2014 companion publication on Designing interactive systems*. ACM, 2014.

Harriman, J. "Pd Poems and Teaching Tools". Proceedings of the 15th International Conference on New Interfaces for Musical Expression. NIME, 2015.

Harriman, Jiffer, Locky Casey, Linden Melvin, and Michael Repper. "Quadrofeelia-A New Instrument for Sliding into Notes." Proceedings of the 11th International Conference on New Interfaces for Musical Expression. 2011.

Hmelo, Cindy E., and Mark Guzdial. "Of black and glass boxes: Scaffolding for doing and learning." *Proceedings of the 1996 international conference on Learning sciences*. International Society of the Learning Sciences, 1996.

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

Jordà, Sergi, et al. "The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces." *Proceedings of the 1st international conference on Tangible and embedded interaction*. ACM, 2007.

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

Kapur, A., Eigenfeldt, A., Bahn, C., & Schloss, W. A. (2009). Collaborative Composition For Musical Robots. *Journal of Science and Technology of the Arts*, *1*(1), 48–52. http://doi.org/10.7559/citarj.v1i1.12

Kapur, A. & Trimpin, E. (2007). A COMPARISON OF SOLENOID-BASED STRATEGIES For Robotic Drumming. *ICMC, Copenhagen.*

Kapur, A., & Murphy, J. (2011). The KarmetiK NotomotoN : A New Breed of Musical Robot for Teaching and Performance, (June), 228–231.

Kay, Alan. "Squeak etoys, children & learning." online article, (2005).

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

The Digital Computer as a Musical Instrument. Science 01 Nov 1963: Vol. 142, Issue 3592, pp. 553-557. DOI: 10.1126/science.142.3592.553. M. V. Mathews.

McElhone, Kevin. *Mechanical music*. Vol. 333. Osprey Publishing, 2004.

McLean, Alex, and Geraint Wiggins. "Bricolage programming in the creative arts." *22nd Psychology of Programming Interest Group* (2010): 30.

McPherson, Andrew. "TouchKeys: Capacitive Multi-Touch Sensing on a Physical Keyboard." Proceedings of the International Conference on New Interfaces for Musical Expression, Ann Arbor, Michigan. 2012.

McPherson, Andrew. "The magnetic resonator piano: Electronic augmentation of an acoustic grand piano." *Journal of New Music Research* 39.3 (2010): 189-202.

Miranda, Eduardo Reck, and Marcelo M. Wanderley. *New digital musical instruments: control and interaction beyond the keyboard*. Vol. 21. AR Editions, Inc., 2006.

Murphy, Jim, Ajay Kapur, and Dale Carnegie. "Better drumming through calibration: Techniques for pre-performance robotic percussion optimization". Proceedings of the International Conference on New Interfaces for Musical Expression, Ann Arbor, Michigan. 2012.

Murphy, Jim, Ajay Kapur, and Dale Carnegie. "Musical robotics in a loudspeaker world: Developments in alternative approaches to localization and spatialization." *Leonardo Music Journal* 22 (2012): 41-48.

Oh, Hyunjoo, Jiffer Harriman, Abhishek Narula & Mark D. Gross. "Crafting Mechatronic Percussion with Everyday Materials." Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction. ACM, 2016.

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.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceeding of the 2005 Conference on Interaction Design and Children - IDC '05*, 117–122. http://doi.org/10.1145/1109540.1109556

Resnick, M., Berg, R., & Eisenberg, M. (2000). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. *Journal of the Learning Sciences*, *9*(1), 7–30. http://doi.org/10.1207/s15327809jls0901_3

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. doi:10.1145/1518701.2167142

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.

Reiser, B, Tabak, I. "Scaffolding." *The Cambridge handbook of the learning sciences. Second Edition* (2015)

Shneiderman, Ben. "Creativity support tools: A grand challenge for HCI researchers." *Engineering the User Interface*. Springer London, 2009. 1-9.

Trappe, Christoph. "Making Sound Synthesis Accessible for Children." *Proceedings of the 12th International Conference on New Interfaces for Musical Expression*. 2012.

Turkle, Sherry, and Seymour Papert. "Epistemological pluralism: Styles and voices within the computer culture." *Signs* 16.1 (1990): 128-157.

Leitman, Sasha. "Trimpin: An Interview." *Computer Music Journal* 35.4 (2011): 12-27.

Long, Jason, et al. "A methodology for evaluating robotic striking mechanisms for musical contexts." *International Conference on New Interfaces for Musical Expression*. 2015.

Overholt, D. (2011). The Overtone Fiddle: an actuated acoustic instrument. *Proceedings of the New Interfaces for Musical Expression Conference. 2011*, (June), 30–33.

Rogers, T. Kemper, S. Barton, S "MARIE: Monochord-Aerophone Robotic Instrument Ensemble" *Proceedings of the 15th international conference on New interfaces for musical expression*. ACM, 2015.

Peppler, K., & Kafai, Y. (2005). Creative coding: Programming for personal expression. Retrieved August, 1–7.

Petrich, M., Wilkinson, K., Bevan, B. (2013). It Looks Like Fun, But are they Learning? In M. Honey & D. Kanter (Eds.), Design, make, play: Growing the next generation of STEM innovators (pp. 50 - 70). New York: Routledge.

Shanken, Edward. "New Media, Art-Science, and Contemporary Art: Towards a Hybrid Discourse?." Artnodes 11. 2011.

Rusk, Natalie, et al. "New pathways into robotics: Strategies for broadening participation." *Journal of Science Education and Technology* 17.1 (2008): 59-69.

Shear, Greg, and Matthew Wright. "The Electromagnetically Sustained Rhodes Piano." *Proceedings of the 11th Conference on New Interfaces for Musical Expression*. 2011.

Sheffield, E., & Gurevich, M. Distributed Mechanical Actuation of Percussion Instruments, 11–15. Proceedings of the 15th Conference on New Interfaces for Musical Expression, 2015.

Singer, E., Feddersen, J., Redmon, C., & Bowen, B. (2004). LEMUR ' s Musical Robots. *Proceedings of the 4th Conference on New Interfaces for Musical Expression*, (2004).

Verplank, Bill, and Francesco Georg. "Can Haptics Make New Music?-Fader and Plank Demos." *NIME*. 2011.

Weinberg, G., & Driscoll, S. (2006). Toward Robotic Musicianship. *Computer Music Journal*, *30*(4), 28–45. http://doi.org/10.1162/comj.2006.30.4.28

Vossoughi, Shirin, and Bronwyn Bevan. "Making and tinkering: A review of the literature." *National Research Council Committee on Out of School Time STEM* (2014): 1-55.

Webb, David C., Alexander Repenning, and Kyu Han Koh. "Toward an emergent theory of broadening participation in computer science education."*Proceedings of the 43rd ACM technical symposium on Computer Science Education*. ACM, 2012.

Zimmerman, John, Jodi Forlizzi, and Shelley Evenson. "Research through design as a method for interaction design research in HCI." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007.