The Shake Stick

Alex Wilson[1] and Abram Hindle[2]

[1]**Department of Computing Science, University of Alberta, Edmonton, Canada, aewilson@ualberta.ca,**
[2]**Department of Computing Science, University of Alberta, Edmonton, Canada, hindle1@ualberta.ca**

ABSTRACT

We present a new embedded instrument, with discussion on the challenges of developing embedded instruments, and the practice and theory of NIME evaluation and design. The Shake Stick is a Raspberry Pi-based embedded instrument using SuperCollider for granular synthesis. In our analysis and design, we explore the MINUET design framework, dimension space analysis for inter-instrument comparison, and learning curves. Furthermore, we discuss lessons learned from using the instrument in group improvisation, as well as challenges and prospects for the creation of sound palettes used in the granular synthesis.

Keywords: NIME, HCI

1 INTRODUCTION

Although there is a variety of work being done in the NIME field on design methodologies, there is no consensus on the application of this work to the design of new instruments. The authors of this paper present their instrument and the design process and evaluation methodologies that were used in its development. In addition to demonstrating the practicality of the chosen frameworks, we also offer some comment on where and how they were useful to us as practitioners.

The Shake Stick is a granular synthesis-based instrument with simple gestural control. The instrument creates grains out of a buffer that is loaded from a file on the device, with the player in control of some granularization parameters. For the most part, this control is created with a simple mapping of roll/pitch/yaw of the device. Grains always have a Hanning envelope. The player can control the grain starting position within the buffer by changing the roll, and grain length is mapped to pitch. In addition, multiple mappings have been tested for dynamics control. Grain position is varied across the entire buffer, while grain length can be varied within a range empirically determined to produce interesting sounds.

The contributions of this paper are:

- An open-source, low-cost NIME
- A description of the design and analysis process of the instrument
- A case-study and discussion of several design tools proposed by the NIME community

2 METHODOLOGY

After an initial literature review, design proceeded according to the MINUET framework, with some additional analysis and goals coming from literature on dimensionality in NIMEs and learning/appropriation. MINUET divides the design into two phases, Goal, in which the objectives and requirements of the design are formalized, and Specifications, in which technological aspects such as mapping, control, and implementation are considered. The creators of the framework insist on a rigid divide between the two, precluding any simultaneous work on goals and specifications. Although this divide seems sensible, in practice it was found to be challenging, as barriers to implementation may naturally effect the desires laid out in the Goal stage.

For instance, in the initial MINUET analysis, one of the desired activities for which the instrument could be used was facilitating unusual interactions with one's own voice. The difficulty of configuring audio input on the Raspberry Pi, however, made achieving this goal less desirable, so no prototype worked in this way and the design was changed.

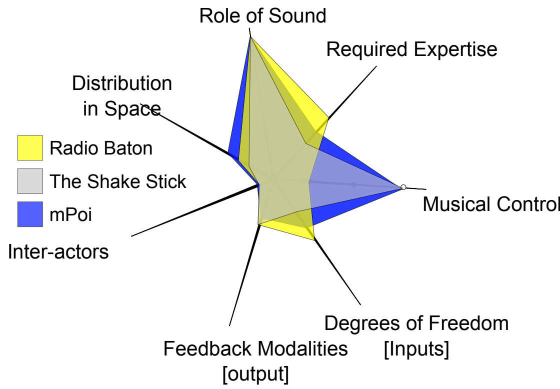

Figure 1. Dimension Space Analysis for three instruments: Radio Baton, The Shake Stick, and mPoi.

Ultimately, the MINUET analysis proved useful in creating a teleological foundation for the design, and the goals did not change too much over time. One of the goals in the creation of MINUET is "to guide the evaluation process." Indeed, having a set of goals written down throughout the design/evaluation/iteration process was very helpful. The framework on its own, however does not provide any specific methods of evaluation. Here, the authors turned to other works...

Considering the rapid proliferation of NIMEs, any new device will be compared to an ever-increasing number of NIMEs. Dimension space analysis, as explored by Birnbaum et al. (2005), can facilitate communication, comparison, and evaluation of NIMEs. In figure 1, we have provided a dimension space analysis of our NIME using the methodology of Birnbaum et al. (2005). In addition, we have included an analysis of two other instruments which influenced the design of the Shake Stick. This analysis was helpful to the authors in providing a mental model for quickly comparing our evolving design with past iterations as well as the design of other instruments.

The design of our NIME, although it is intended to be somewhat novel, was influenced by several other instruments. From the Radio BatonMathews (1991), we draw upon a simple vocabulary of movements, but most importantly, a focus not on controlling the pitch of sound produced, but enabling expressiveness through by controlling other aspects of the music.

The mPoiNam (2013) provided some inspiration in terms of an interaction model. One of the initial ideas for the design of our NIME included a Poi-like instrument-on-a-string design. Although this design was not carried through into prototype or instrument, enabling circular swinging motions, which the authors note provides a joyful experience to players, remained a soft goal throughout the design.

Although enabling ownership and appropriation were not always at the forefront in the design process, the possibility of appropriation and experimentation was nonetheless considered important throughout the design. According to Zappi and McPherson (2014), the possibility of appropriation is aided by creating a NIME with low degrees of freedom. In constraining players, it appears, they are more likely to engage creatively with the instrument. This work, like the work of Mathews with the Radio Baton, enforced a desire to move away from pitch-based control to foster an expressive and interesting experience.

3 IMPLEMENTATION

The device, which can be seen in figure 2, consists of the following hardware:

- A Raspberry Pi B+[1], with AdaFruit case and GPIO cable
- A paint-stirring stick from Home Depot
- An AdaFruit LSM9DS0 Sensor Board[2]
- An inexpensive USB Sound Card
- A portable 2200mAh, 1A, USB Li-ion battery[3]
- A potentiometer for volume control

[1]Raspberry Pi B+ http://www.raspberrypi.org/products/model-b-plus/
[2]LSM9DS0 product page http://www.adafruit.com/product/2021
[3]USB Li-ion battery product page http://www.adafruit.com/products/1959

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.995v1 | CC-BY 4.0 Open Access | rec: 21 Apr 2015, publ: 21 Apr 2015

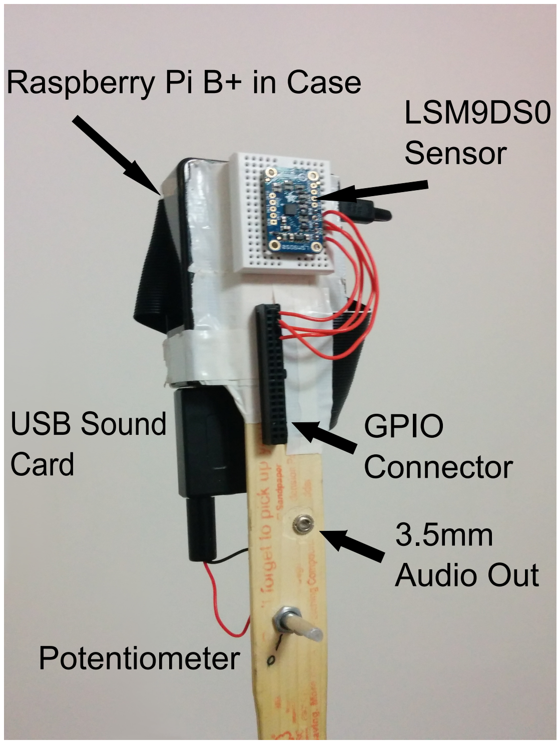

Figure 2. The Instrument

- A 3.5mm mono audio jack for audio output
- A portable speaker with Li-ion

All together, the hardware for this device costs about $150 CAD. With fully-charged batteries, the authors have successfully run the device and speakers for over 3 hours at a time without issue.

The device runs Satellite CCRMABerdahl et al. (2013) Linux distribution, as well as the newest versions of jackd and SuperCollider. The CCRMA system was created with the intent of easing the development of embedded musical devices, as well as fostering ease of use and longevity in the resulting instruments. As such, it is designed to facilitate communication with and free-standing operation of the device. For instance, it is very easy to configure a device running Satellite CCRMA with a default synthesis program that will run when the device is started, with no performer intervention necessary. Our NIME deviates from the usual Satellite CCRMA setup in several ways:

- There is no Arduino present; all sensor interaction is achieved via the Raspberry Pi's GPIO interface.

```
s.waitForBoot({
    Buffer.readChannel(s, "/home/ccrma/git/nime/data/audio.wav",
      channels:[0],
      action:  { |b|
          {
            // granulate out of the buffer at a position based
            // on roll, with length of grains based on pitch
            GrainBuf.ar(1, Impulse.kr(20), AHRS.pitchKr(0.5/360, 0.60),
                sndbuf:b, pos:AHRS.rollKr(1/360, 0.5));
          }.play;
    });
}, 300, {"oops".postln});
```

Figure 3. SuperCollider development with custom UGens.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.995v1 | CC-BY 4.0 Open Access | rec: 21 Apr 2015, publ: 21 Apr 2015

- Although they support using a Raspberry Pi, Satellite CCRMA is geared towards using a BeagleBone instead.
- The Satellite CCRMA creators suggest using Pure Data Extended, whereas we used SuperCollider

Despite these deviations, using the pieces of Satellite CCRMA that we did was still helpful. In comparison to the Raspbian Linux distribution, there was much less work to do in getting a workable environment for the instrument when using CCRMA. Furthermore, the community that exists around the Satellite CCRMA project has created a large amount of documentation and discussion on their wiki and mailing-list which frequently proved helpful in diagnosing and resolving technical issues.

The LSM9DS0 sensor used in our NIME provides three degrees of freedom sensors for acceleration, rotation, and magnetic fields, in addition to sensing temperature. The sensor interfaces with the Raspberry Pi via I2C. The authors adapted AdaFruit's AHRS (Attitude and Heading Reference System) and LSM9DS0 libraries, originally written for use on the Arduino, to work with the Raspberry Pi. The system interfaces with I2C via a Wiring-like[4] wrapper around the I2C/SMBus interface provided by Linux. Thanks to this existing software, we were able to quickly create a SuperCollider plugin providing UGens for accessing the roll, pitch, and yaw of the device, as determined by the LSM9DS0 sensor. During the course of development, a UGen was also created for reading HIGH/LOW data from the GPIO pins on the raspberry pi. The resulting code is available on github[5], and an example of the SuperCollider code can be seen in 3. Avoiding using the additional component of an Arduino reduces cost and energy consumption. One potential drawback of this method is that the Raspberry Pi does not provide any analog input; for this an external ADC would be required. One issue with the approach of creating SuperCollider plugins was the unreliability of input gathering, which relies on Linux system calls. This was circumvented by having another thread continuously updating the input data, with the UGens providing the newest data available to them as it is requested.

Another important aspect of the instrument is the design of the buffer from which grains are created. Because this buffer is not played directly, but rather provides a set of acoustic possibilities from which the user can draw, we refer to it as a palette. The palettes were developed in the Audacity audio editor. An example palette is shown in figure 4. The samples in the palette can be drawn from any musical source, and greatly influence the sound of the instrument. For instance, by including drum samples, one can create a percussive sound for the instrument, whereas including speech generally creates a very different feel. Having quiet or silent spots in the palette has been found to be very helpful, as it provides a resting place for the performer, and also allows one to gradually ease in to adjacent sounds.

The possibilities for these sound palettes are endless, and for each sound palette created, the instrument regains a necessity of exploration. Anecdotally, we have experienced divergent and convergent phases of navigation when using the Shake Stick, much like those discussed by Tubb and Dixon (2014). The consideration of palette creation as part of the instrument can complicate, in some ways, the evaluation of the instrument. For instance, Jord'a (2004) examines the efficiency of instruments over time for individual players, concluding that, for instance, the Kazzoo provides a low maximum efficiency that is achieved very quickly, while the Piano has a high efficiency which is achieved after years of practice. The Shake Stick, with a fixed palette, might have a learning curve (efficiency over time) somewhere between the Kazoo and the Kalimba. It seems to provide an exciting period of growth that levels off relatively quickly. With the addition of palette creation and editing, however, there is a renewed change in efficiency, and with each new palette comes another period of learning how to play.

Although other parts of the instrument are important, the potential for transforming the quality of sound, and even player enjoyment by modifying the sound palette suggests that this paradigm may be a very powerful way to create ownership and appropriation in low-dimensional devices. Once again, we can draw a comparison to the Radio Baton, where a piece of music, as performed with either instrument, will be created both on-stage and off. The palette, or the score and conductor program, are developed and tweaked before the performance, and are then brought to life by the performer. As in the case of the Radio Baton, this preparation does not need to harm liveness or expressiveness.

[4]Arduino Wiring library reference http://arduino.cc/en/Reference/Wire
[5]Shake Stick source code on github https://github.com/yourpalal/lsmd9dso-supercollider

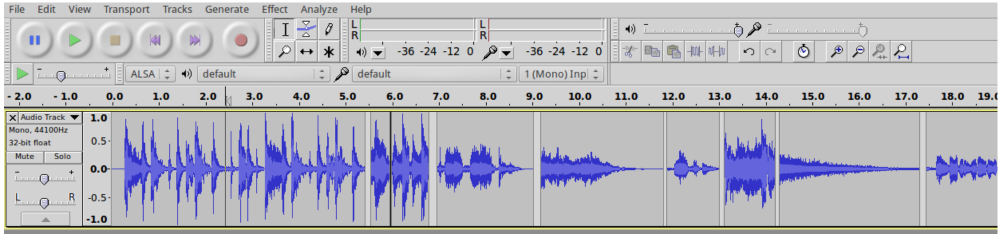

Figure 4. Sound Palette Creation in Audacity

4 DISCUSSION

The process of designing and implementing the Shake Stick provided an opportunity for exploring, in a broad sense, the field of NIME design and evaluation. More concretely, this instrument provided challenges and surprises along the way which may be useful for other practitioners in this field. For instance, frequent use of the instrument throughout its multiple iterations was very helpful in understanding the capabilities and affordances of the instrument, as well as helping uncover a few surprises. When testing an early prototype with an experimental improvisational group, the near-total lack of control over dynamics, which may not be a problem in a solo improvisational setting, created difficulties in trying to blend with other musicians. In this same session, it became apparent that one of the components used in the design, a momentary button, created an audible click which detracted from the expressiveness of the instrument. The original Goal analysis within the MINUET framework was not concerned with this sort of group improvistation, and it quickly became clear during practice that the fit of the Shake Stick with this activity and context could be greatly improved. At this point, we added new requirements to the design, and produced an instrument which felt much more capable in these situations. Initially, this meant removing the loud button (which provided mute/unmute funcitonality) and adding an audio potentiometer to the audio output, thereby providing rudimentary dynamics control. This provided an surprising change to the feel of the instrument. Although the mute/unmute button was originally added as a simple way of ensuring that the instrument could be moved without producing unwanted racket, and the potentiometer could theoretically accomplish the same thing, the instantaneous nature of the button proved to be very engaging, allowing for more percussive styles of play. The potentiometer, on the other hand, is better for gradual, continuous dynamics.

Another lesson learned is that the flexibility of Linux and SuperCollider is very convenient, allowing instruments and mappings to be quickly reconfigured, debugged, and broken down into their constituent parts for analysis. For instance, early prototypes produced rather unexciting sinusoidal output, which allowed for some level of independence in hardware, mapping, and synthesis. Similarly, the basis for some mappings were developed and tested on a laptop running Linux and SuperCollider, with mouse coordinates as a stand-in for AHRS data.

4.1 Future Work

As development has progressed on the Shake Stick, various deficiencies and opportunities have become apparent. Ideas we would like to pursue include:

- providing a custom, more interactive method of palette creation and editing
- adding a networked component, so that changes to the palette could be introduced by other performers or the audience
- adding support for multiple palettes that can be chosen in real-time, which could enable more dynamic performances
- adding a monitor for the audio so that the performer could listen to the potential output of the device, exploring the space without having to expose the audience to the variety of interstitial sounds this can produce
- investigating porting over the instrument to an Android phone, most of which include a large variety of sensors, equivalent or greater computational power, built-in batteries, as well as networking and display capabilities. This may reduce cost even further, and allow for performances including multiple instances of the Shake Stick (Shake Phone?) at a time.

- investigating the effect on convergent/divergent behaviour of more complicated mappings of space to position, such as the space-filling curves explored by Tubb and Dixon (2014).

REFERENCES

Berdahl, E., Salazar, S., and Borins, M. (2013). Embedded networking and hardware-accelerated graphics with satellite ccrma. In Yeo, W., Lee, K., Sigman, A., H., J., and Wakefield, G., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 325–330, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST.

Birnbaum, D., Fiebrink, R., Malloch, J., and Wanderley, M. M. (2005). Towards a dimension space for musical devices. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 192–195. National University of Singapore.

Jord'a, S. (2004). Digital instruments and players: Part i – efficiency and apprenticeship. In Nagashima, Y., Ito, Y., and Furuta, Y., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 59–63, Hamamatsu, Japan.

Mathews, M. V. (1991). The radio baton and conductor program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal*, pages 37–46.

Nam, S. (2013). Musical poi (mpoi). In Yeo, W., Lee, K., Sigman, A., H., J., and Wakefield, G., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 148–151, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST.

Tubb, R. and Dixon, S. (2014). The divergent interface: Supporting creative exploration of parameter spaces. In Caramiaux, B., Tahiroglu, K., Fiebrink, R., and Tanaka, A., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 227–232, London, United Kingdom. Goldsmiths, University of London.

Zappi, V. and McPherson, A. (2014). Dimensionality and appropriation in digital musical instrument design. In Caramiaux, B., Tahiroglu, K., Fiebrink, R., and Tanaka, A., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 455–460, London, United Kingdom. Goldsmiths, University of London.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.995v1 | CC-BY 4.0 Open Access | rec: 21 Apr 2015, publ: 21 Apr 2015