Re-Designing Principles for Computer Music Controllers: a Case Study of SqueezeVox Maggie

Perry R. Cook
Princeton University Computer Science (also Music)
35 Olden St. Princeton, NJ USA 08540
prc@cs.princeton.edu

Abstract
This paper revisits/extends "Principles for Designing Computer Music Controllers" (NIME 2001), subsequently updated in a NIME 2007 keynote address. A redesign of SqueezeVox Maggie (a reoccurring NIME character) is used as an example of which principles have held fast over the years, and which have changed due to advances in technology. A few new principles are also added to the list.

Keywords: HCI, Composed Instruments, Voice Synthesis, Wireless, Batteries, Laptop Orchestras, SenSAs.

1. The Original Principles

To begin, I will restate the original 13 Principles for Designing Computer Music Controllers [1], briefly noting whether they still hold true (or not) and why. Two new principles will be introduced here, and more added later.

1.1 Some Human/Artistic Principles

1) Programmability is a curse
Still true. The ability to constantly add features means that many devices are never used in performance or formally tested. Ever-changing NIMEs make it hard to determine, especially after time, if they still work as intended.

2) Smart instruments are often not smart
Still (largely) true. Instruments that learn from the player can easily confuse (everyone). However, the last decade has seen major advances in machine learning and music information retrieval [2]. Cautious use of these techniques can allow crafting of systems that learn. To be most useful, it is still best to let the human player/designer explicitly choose training vs. performance modes.

3) Copying an instrument is dumb, leveraging expert technique is smart
Still true. The best violin is a violin, but we've seen many great virtuosity-inspired controllers: violins, trumpets, mic stands, turntables, beat-boxers, ethnic instruments, etc.

4) Some players have spare bandwidth, some do not
Still true. Trumpets still have 3 valves and trumpet players still have 10 fingers. Singers and dancers have lots of spare digits and other manipulators, clarinet players less so, organists even less.

5) Make a piece, not an instrument or controller
This is still true, especially for beginning designers, but even for experienced builders starting a new project. But for mature instruments, tested during many performances, it is possible to redesign to make a NIME more generically useful (see Maggie's redesign below), but we should try to:

18) NEW: Redesign with backward compatibility
More than just an exercise, "backward compatibility" for original functions and compositions can help detect lost functionality, and map progress (more on this shortly).

6) Instant music, subtlety later
Still true. It is compelling for a musical instrument to make sound immediately, simply and reliably. There are other, non-musical reasons that this is a good idea, such as:

19) NEW: Design (and pack) for post-9/11 travel
I have been quite ama(u)zed at my experiences with airport security. More than once the person searching my carry-on bag (after seeing odd things in the scanner) has seen a 5-pin DIN (MIDI) jack, said "Oh, music," and sent me on my way. I also try to pack a copy of a paper depicting and describing the instrument. When asked to prove what the thing does, being able to plug it in and make music often eases the need for more questions. Even if it's a rubber chicken with a MIDI jack, life is easier if it can make some sound pretty quickly. Making security wait while we grumble about MAX or ChucK doesn't help keep us safe from the terrorists. (More on this in Maggie's redesign).

1.2 Some Technological Principles

7) MIDI = Miracle, Industry Designed, (In)adequate
Still true. MIDI is still here, and we still use it. But Open Sound Control (OSC) has become a great option too.

8) Batteries, Die (a command, not an observation)
Not so true anymore. Digital cameras, cell phones, MP3 players, and other small devices have brought smaller batteries with higher energy densities, capable of powering our systems for hours. Cyber-instrument builders can now reasonably consider designing with batteries as an option.

9) Wires are not that bad (compared to wireless)
This also is not so true anymore. Indeed, I would not write this paper if some things had not changed in major ways in a decade. We can now consider building our instruments to use Bluetooth, 802.11 (WiFi), Zigbee and roll-your-own radios. We can also exploit sensor-rich wireless products like the WiiMote, iPhone, and other devices for musical purposes. Wires are still nice though.

1.3 Some Other Principles

10) New algorithms suggest new controllers
11) New controllers suggest new algorithms
12) Existing instruments suggest new controllers
13) Everyday objects suggest amusing controllers

All still very much true, as proved at conferences such as NIME, ICMC, DAFX, SIGGRAPH, etc., and also in the market place. Controllers inspired by new algorithms, and algorithms emerging from new controllers demonstrate the magic that can happen when acoustics, mathematics, physics, haptics, sensors, and many other disciplines inform the design of a new instrument or system. A wide variety of wacky yet quite musical devices ranging from my CoffeeMug and Romaineraca to Eric Singer's Sonic Banana and Musical Raddichio, and many other such "gag NIMEs" have proven to be so memorable that I feel that I can offer the following axiom:

13b) Funny is often much better than serious.

This possibly reflects my personality, but a post-concert review of "What the heck was that? Did you see the dude playing the lettuce?" is preferable to "Wow. That really sucked. What the heck was that?"

2. Some (more) New Principles (NIME07)

My first revisit of the original principles came about when I was asked to give a NIME07 keynote talk at NYU. That talk, entitled "Principles for Controlling Computer Music Designers"[3] presented a survey of the original principles, with modifications as noted above, and also reviewed my experiences from teaching NIME-type courses for two decades. From our experience with the Princeton Laptop Orchestra (PLOrk)[4][5], I added three new principles:

14) More can be better! (but hard).

Years of building musical controllers and speaker systems, and playing in small ensembles of computer-mediated instruments led Dan Trueman and I to ask, "what might it be like to build a large ensemble of such instruments?" The answer began to unfold with the creation of PLOrk in 2005 (Figure 1), and we've been looking at related questions ever since. It's been lots of work, but incredible things have happened and it's been well worth the trouble.

15) Music+Science is a great teaching/ marketing tool

Many of us find ourselves nearly constantly raising money, and explaining to others why what we do is scientific, interesting, difficult, and important. Our field increasingly bridges disciplines and tests the boundaries of computer science, psychology, engineering, physiology, and others. The real-time latency constraints of sensor acquisition, DSP, and interactive music synthesis are tighter in our applications than in any almost all other engineering areas.

The press loves a good nerd-art story (Figure 2), even if they don't actually "get it," and granting agencies increasingly like the idea of funding ground-breaking multimedia research. From a pedagogical perspective, teaching computer science is more fun when the results can be heard and felt immediately. Engineering students dig fun examples that demonstrate key-concepts from their field, while students in the arts and humanities feel technically empowered, often times learning things that they never would have voluntarily signed up for.

Figure 1. The first PLOrk.

MUSIC

But Where Do You Put The Capo on a Laptop?

By BRIAN WISE

PRINCETON

As if keeping your own computer from crashing wasn't hard enough, imagine trying to coordinate 15 laptops during a live musical performance, each one containing experimental software and carrying out its own highly specific musical function.

Engineering such performances is exactly what the Princeton University faculty members Dan Trueman, a composer and violinist, and Perry Cook, director of the school's sound lab, do with the Princeton Laptop Orchestra, an ensemble made up of students using networked laptops connected to small multidirectional speakers and a variety of control devices. The group will perform at the Chancellor Green Rotunda in Princeton on May 2.

"It's an unbelievable challenge to compose for this," Mr. Trueman said. "We've

Laura Pedrick for The New York Times

Boot me up: the Princeton Laptop Orchestra at its debut concert.

Figure 2 PLOrk press.

16) The younger the student, the more fearless

All of us who teach a NIME/PLOrk or related course have experienced the odd fact that the things our students think should be easy to do as projects often are the truly difficult open problems in our field, while the things they think are impossible have been solved many years ago. A great aspect of this is that we (and they) are often proven wrong. In the first PLOrk (a freshman seminar), none of us knew what to expect, or whether it was going to work at all. Things we thought should work never did, and many "impossible" problems worked fine with little effort. This year a few of our fresh-persons are returning to PLOrk one more time as seniors. Others have made a career of PLOrk, taking it multiple times, learning more, and teaching others. We learned a lot, but often feel like we still know very little. We head into each new season, semester, rehearsal, and concert with anticipation, but also white-knuckle panic at possible impending train-wrecks.

One common theme of the three new principles **14-16** is that our NIME field, and our instruments, might be maturing. This leads to the third main topic of this paper:

3. Principles for Controller (Re)design

It seems like more than eight years has passed since the first NIME, because so much has changed. I've modified my stance on principles regarding batteries and wireless/ wires, and relaxed my position a little on smart instruments that learn from the player. I also mentioned in my revisit of Principle **5 (Make a piece, not an instrument or controller)**, that after a NIME has been tested extensively, and used in a variety of demos and musical performances, it is reasonable to think about a redesign. So here is one:

3.1 SqueezeVox Maggie as a Case Study in Redesign

Building many similar controllers and gigging quite a lot on them since the SqueezeVoxen were first created [6] [7], I formed many opinions about issues like functions and integration (speakers, possibly amplifiers, and possibly even all computation contained within the device), portability (accordion vs. concertina), and expressive control (pitch control via AGO keyboard vs. brass-valve-mapped buttons vs. Boehm-fingering-mapping of buttons vs. linear FSRs vs. tilt sensors etc.). I entered a phase where I wanted to craft a NIME to constantly carry with me; to build an "Axe" that would be my main instrument for a while; on which I could compose and perform many pieces, and play pieces by and with others (somewhat in opposition to **Principle 5)**). Maggie seemed to be the obvious choice, because she looks and feels like a real instrument, and nestles nicely into a backpack.

To begin the redesign, and even for initial designs:

(20) NEW: Build a (new) copy, don't trash the original
Once one undertakes a redesign, the strong temptation is to modify the original controller, but we should try to resist this. Keeping the original around allows for verification and before/after comparisons. Maggie's redesign clone copy is named Milhouse. If we modify the original we also risk breaking it, and having no reference for improvements or functional testing. A corollary is:

(20b) NEW: Build two or more if you can afford it.
For the construction of a new controller, build more than one, because inevitably they come out differently. Best case, one will come out better. Worst case one won't work at all and the functional one can be used to determine what's broken in the other. I do this for nearly everything, including kits. If I need two RF modules, I order three kits because one doesn't work or somehow gets fried.

Another corollary of principle **20)** is **18)**: **Redesign with backward compatibility.** Maintaining all functions and musical pieces the original device supported allows comparisons to be made and improvements to be mapped. Lost functions or capabilities can also be tracked. It is possible; even likely, that each device might have attributes that make it better than the other for certain functions and pieces. This of course might even suggest a further redesign, merging the best parts of both. In the original Maggie, bellows pressure combined with the four right-hand buttons (brass valve pitch mapping) and thumb slider (overtone mapping) to send standard General MIDI NoteOn and AfterTouch. I thought of ditching this arcane functionality, but it fits so well with the **Instant Music Principle 6)** and **Post-9/11 Principle 19)** (plug her into any MIDI synth for instant music) that I had to leave it in.

3.2 So what's new about Maggie for 2009?

Figures 3 and 4 show Maggie's main control surfaces. As in the original SqueezeVox Maggie, the left hand side sports an array of 32 momentary push buttons (1) sending 32 different values of MIDI control #6 as well as NoteOn messages for 32 standard MIDI drums on channel 10, and the bellows vent-switch (2), which still performs the function of allowing air to enter or leave the bellows rapidly, but also sends state on MIDI control #65. The toggle "bank switch" for mapping the 32 buttons into 64 effective buttons (compatible with SqueezeVox Lisa's 64 buttons) was basically never used in practice, and my left hand had to de-strap to flip the switch anyway. So in Maggie09, there are two arbitrary bank & mode toggle push buttons (3,4) that transmit state on MIDI controllers #67 & #68. Purple and green LEDs are located on top (5, not visible) to indicate the state of the mode/bank switches.

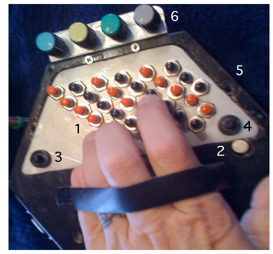

Figure 3 Maggie's left side.

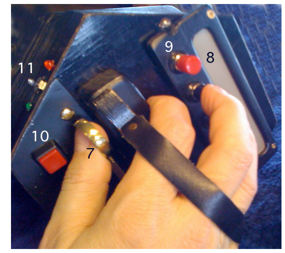

Figure 4 Maggie's right side.

The four bend sensors previously located on the top of Maggie's left side turned out to not be so useful, and often broke and needed repair. Further, Maggie lacked any variable analog sensors capable of being positioned to a static value (potentiometers), and I had gotten accustomed to having these for effects, tempos, etc. in the VOMID [7]. So the bendies were replaced with four pots that report their values individually as MIDI controllers #40-43 (6).

Maggie's right side still has a slide pot with thumb-ring (7) (MIDI controller #1), the linear FSR strip (8) (MIDI controller #4), and four "valve" buttons (9) mapped to brass fingerings (more on that shortly). The first button springs were so strong that they made my fingers hurt, so new buttons with longer action throws and weaker springs were installed. The new switches also each report their states as MIDI controllers #80-83. The right side also has a push-button toggle switch (10) (MIDI control #64), (same as SqueezeVox Lisa's accordion register switch). Status of this is displayed the top right side (11) via a blue LED. There's also a power switch and red power LED.

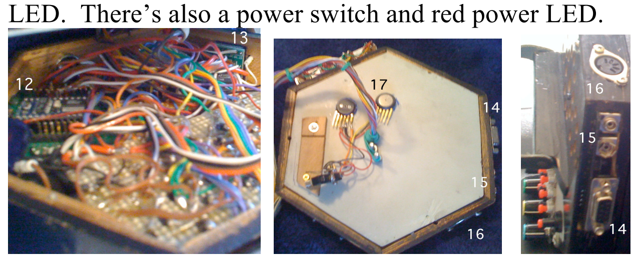

Figure 5 Left inside computer guts, sensors, and connectors.

Figure 5 shows Maggie's left insides and connectors. All computer functions are located in the left endcap (12), and there is a multi-color wiring harness running through her bellows to connect her left and right sides, terminated with a DB25 connector on one end for easy disassembly. This leads us to a new design (and redesign) principle:

(20) NEW: Wire and document for future surgeries.
When I take an interface apart to add functions, replace a switch, LED, or whatever, one or more wires often break. The good news is that I usually wire my interfaces using a different wire color for each sensor, display, or function. I also write with a sharpie or pencil next to most major connections a letter representing the wire color that should be there. This makes it easy to fix, replace, or copy things.

For her 2009 internal makeover, Maggie received a pair of more sensitive Motorola MPX5050 air pressure sensors (17). These combine in MIDI control #66 to indicate whether the bellows are blowing (inward pressure) or sucking (outward pressure), and the absolute value of blow/suck pressure is carried by both MIDI control #7 and MIDI after-touch. Maggie's previous left-hand bendies had been intended for control of vowel-space (formants, F1/F2, articulation, etc.) and other multi-dimensional data, but in the redesign I took a lesson from Lisa's brother Bart (by Colby Leider) [6] and embedded a two-axis accelerometer (ADXL202) inside her left top plate (13). Maggie still has an RS232 jack (14) (to talk to the microprocessor), a 9V battery clip (15), and a MIDI (16) jack on her left side.

3.3 What's Radical and Awesome About Maggie09?

Internal battery power and wireless MIDI. There, I said it. After Maggie's external battery fell off a couple of times during gigs, I installed eight 1.2V 1.5 Amp Hour Ni-MH rechargeable phone batteries inside her right end-cap (Figure 6). As I roam freely, Maggie-Nouveau transmits MIDI via an M-Audio MidAir (Figure 7) system.

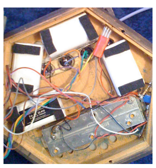

Figure 6 Internal batteries. Figure 7 MidAir wireless MIDI.

Transmitting MIDI to what? An integrated "Maggie wrangler" written in ChucK, with GUI "Dashboard" built in MAUI (MiniAudicle User Interface), as shown in Figure 8. This motivates two final new principles:

(21) NEW: Build diagnostic features and displays

One problem with sensor-rich controllers is that it's hard to tell if things are working, and the audience often figures out that things are broken just about when we do. Visual feedback, even if just for setup and debugging, can save hours. Nice meaningful feedback, like the simple LEDs mounted on Maggie's top, and her elaborate GUI dashboard, can help us build, repair, set up, and perform.

(22) NEW: Construct controller proxies

Creating even a minimal GUI that sends the same signals as the controller can be helpful in many ways. Part of this new principle is a corollary of principle **19) Design for travel in a post 9/11 world**. The GUI of Figure 8 is fully functional even without Maggie present and connected, and can be used on an airplane or anywhere else to test and tweak new ideas for compositions. Shreds for handling MIDI and MAUI are independent, taking over from each differently depending on whether Maggie is connected or not. To start a new piece or construct a new set of mappings, all I need to do is copy the dashboard code and start hacking in the main (empty by default) synth shred. I can do this at any time, with or without the controller.

"Proxy-ism" and compatibility work two ways; any decent programmable MIDI keyboard can send almost all of the signals that Maggie's ChucK/MAUI GUI and software understand. Further, if you inspect the MIDI messages I've selected for Maggie to send, you'll note that most are standard: buttons send pitches or drum NoteOns, x/y tilts send MIDI Pan/Balance, vent/bank/mode/register buttons act like MIDI Sustain, Portamento, Sostenuto, and Legato Pedals. More possibility for instant music!

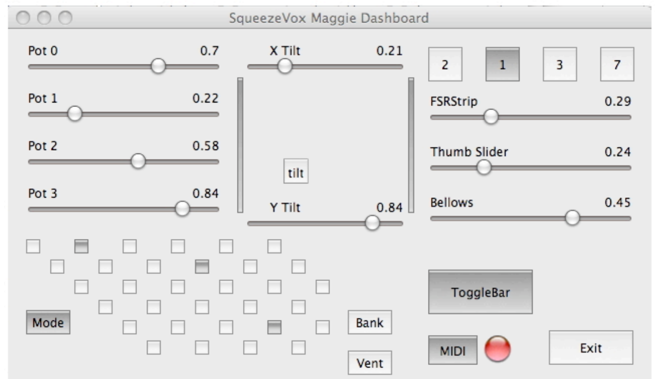

Figure 8 Maggie's massive MAUI GUI display dashboard.

4. Conclusions and Future Work

From the four or so gigs I've done so far with the redesigned Maggie, she works really well. I carry her with me pretty much everywhere. Will I build more interfaces? Yes. Will I constantly revisit these principles, adding new ones, long into the future? Likely. Mostly for now I just want to play my new axe and make lots of music.

5. Acknowledgments

This work was supported by Princeton's David A. Gardner '69 Magic Fund (Humanities Council), and the Kimberly and Frank H. Moss '71 Research Innovation Fund (SEAS). Thanks to NIME, PLOrk, TeQWire, LAP, Colby Leider, Dan Trueman, Ajay Kapur, Curtis Bahn, and Ge Wang.

References

[1] P. Cook "Principles for Designing Computer Music Controllers," ACM SIGCHI New Interfaces for Musical Expression (NIME) Workshop, Seattle, 2001.

[2] International Symposium on Music Information Retrieval, (ISMIR) 2000-2008, www.ismir.net.

[3] P. Cook, "Principles for Controlling Computer Music Designers," Keynote talk, NIME 2007, NYU.

[4] Trueman, D, P. Cook, S. Smallwood, and G. Wang, "PLOrk: The Princeton Laptop Orchestra, Year 1," in *Proc. International Computer Music Conference*, 2006.

[5] Smallwood, S., D. Trueman, P. Cook, and G. Wang. "Composing for Laptop Orchestra," in *Computer Music Journal* 32(1), MIT Press, 2008.

[6] Cook, P. and C. Leider, "SqueezeVox: A New Controller for Vocal Synthesis Models," in Proc. of the ICMC, 2000.

[7] Cook, P. "Real-time Performance Controllers for Synthesized Singing," in Proc. of NIME, Vancouver, 2005.