Escaping the Sandbox: Making and its Future

Tom Jenkins
Georgia Institute of Technology
Digital Media Program
Atlanta, GA, 30332, USA
tom.jenkins@gatech.edu

Ian Bogost
Georgia Institute of Technology
Digital Media Program
Atlanta, GA, 30332, USA
ian.bogost@lmc.gatech.edu

ABSTRACT
"Making" has been gaining traction in HCI and related fields both as a community of practice and as a method for creating objects and systems. While making is an important cultural practice, this paper claims that there is a disconnect between the rhetoric of making and "real world" notions of domain relevance and embedded hardware development. In considering how making operates in practice, we offer the metaphor of a sandbox to describe this contradiction. We exemplify the metaphor with a small-scale prototyping platform of our own, and offer visions on how making might progress in the future.

Author Keywords
Making; design; engineering; embedded hardware; prototyping platforms

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Within many computing communities, "making" is offered as a means to provide deeper, richer learning experiences with computational materials or to help support initiatives around STEAM education [3,6,8]. Meanwhile, outside the academy, making has been amplified by the technology hype machine. The "maker movement" valorizes making for the sake of making, offering Make magazines, books on Making, and Maker Faires to spread the gospel of personal production to anyone who will listen [2].

The most common critique of the maker movement and its associated culture involves the accusation that it exhibits a cyberlibertarian ideology:

The maker movement is born out of, and contributes to, the individualistic, market-based society that has become dominant in our time. ... new technologies promise to create a class of high-tech entrepreneurs ... while allowing them to ignore or simply design their own way around the established political, economic and legal system [16].

It is a reasonable critique of maker culture, but also increasingly indistinguishable from a critique one could levy at all of technology culture today. Indeed, the cyber-libertarian ideology critique might cover over a more mundane but possibly more fundamental issue in maker culture, namely that it is *functionally hermetic* before it is *politically hermetic*. That is, even before making implicitly assumes the cyberlibertarian agenda, it isolates itself from the broader application domains from which it draws inspiration—electrical engineering, home automation, and even ordinary software development, for example.

To illustrate how making works, and what it does and doesn't do, this paper offers the metaphor of a sandbox describing the role of making as a part of broader modes of industrial production; it illustrates this metaphor with a project of our own, in progress; and finally, it offers visions of how the rhetoric of "making" might move forward.

MAKING IN HCI RESEARCH
Making, along with hacking and other DIY practices, has been making inroads among the HCI community in recent years. At CHI 2014, events were even held that modeled themselves after common maker culture practices: a workshop that was itself a hackathon [18]. The ambiguity inherent discussions about making was also reflected in a panel that discussed the ambiguity of making [1].

This current interest around making is rooted in previous work. Kuznetsov and Paulos [7] offered a survey of online fora that support different kinds of DIY communities, positioning the "expert amateur" as creating and supporting knowledge transfer and expertise among groups invested in various modes of creative practice. Rosner and Bean have covered IKEA hacking as a way to express individuality [15] and Rosner and Goodman have taken up other DIY practices like gardening and knitting [4], emphasizing in particular material practices that come alongside personal hand-work.

While we criticize making as a cultural construct in this paper, we don't seek to trivialize the work of those who are doing research with making communities. For example, Tanenbaum et al. [17] take making as a broad category of practice, including ad-hoc design and production as a response to natural disaster. Lindtner has been working on a series of projects focusing on hardware manufacturers that

TEI '15, January 15 - 19 2015, Stanford, CA, USA

ACM 978-1-4503-3305-4/15/01…$15.00
http://dx.doi.org/10.1145/2677199.2680558

identify as makers in Shenzhen, China—particularly the cultural bridges between Western entrepreneurs looking to produce electronics there [10] as well as how these emerging communities understand themselves [9]. Those developing and supporting practices around makers and making are also conducting interesting, worthwhile work. Mellis and Beuchley emphasize the malleability of electronics and its ability to let novices create fully functional, personalized objects as a means of gaining a particular kind of media literacy [11,12,13].

While research in HCI concerned with making is growing, relatively little work attempts to unpack or describe what "making" is or how it operates. Critical making [5,14], deploys making as part of a method to interrogate cultural practices, but it doesn't concern itself with the nature of producing artifacts, but instead the knowledge that the experience of making produces.

It is not easy to discern what "making" might actually *be*. Both scholarly and popular writing on the topic describe such a broad range of activities as to defy being understood as a single practice. Sometimes it seems as if anyone who creates anything whatsoever could be construed as a "maker," and their practices as a kind of "making." It's a deft rhetorical strategy: by construing anything whatsoever as "making," its proponents gain substantial momentum.

The rhetorical cleverness of "making" notwithstanding, in this article we aim to take the concept at face value. We use this definition described in the introduction above to motivate the rest of our discussion.

THE SANDBOX CONFLICT

Think of the community that comprises the maker movement as a sandbox on a lawn. The sandbox is the domain of the makers, distinct from the lawn of "real-world" professional practice. Sand is great in a sandbox, but it only has use and value within the borders of that sandbox. Inside the sandbox, sand is a resource. Outside of it, sand is a nuisance. From the opposite perspective, sand on the lawn doesn't make the lawn a beach; it just kills the grass. Likewise, grass in the sandbox has no function other than as waste.

The Making Sandbox

This metaphor describes how hardware devices like Arduino, Raspberry Pi and other electronics platforms—systems for "making," relate to other kinds of practices. For example, the Raspberry Pi's founders see education as its primary goal. They describe their platform as a "tiny, cheap computer for kids" that was developed to counter a trend among novice computer science students who had little experience with the real material of computing—embedded electronics [19]. Whether or not the Raspberry Pi has been successful at popularizing embedded development, however, is debatable: for much of the device's life (and during the period when the most diverse communities wanted one), actually obtaining the hardware was nearly impossible, becoming a kind of a meta-activity in itself.

The rise of Processing, Wiring, and Arduino came at the same time (and to some extent, were built to service) as the rise of courses and then programs of study in art, design, culture, and computation at NYU, USC, Carnegie Mellon, Georgia Tech, and others. The devices found homes in these programs and then began to spread out into the world after students graduated and took those platforms to other venues in the art, design, and education communities. The primary success of these platforms is not in providing a deep understanding of electronics, but in creating a new ecosystem of tinkering activity—one separate from both electrical engineering and traditional arts practice.

It's no surprise, then, that there are fusions of making and everyday life. The Twine, for example, is a Wi-Fi-operated sensor-based monitoring device [20]. An example application for the Twine shows it placed it in a basement attached to a moisture sensor. It is programmed through a web-based tool, which sends you a text message if your basement floods. At a first glance, the Twine looks like a success for the everyday relevance of making. It is a finished product that has gone on to be sold and used—a proper hybrid of sand and lawn. But further inspection reveals that this isn't quite the case. The Twine is expensive (several hundred dollars with all the sensors), and as such its use as a piece of sensor hardware is at odds with its role as a tinkering platform one can use and reuse through web-based programming—particularly if it is also meant to be put to a permanent, single-purpose use as is suggested. Instead of becoming a disposable ubiquitous computing tool, it is a trinket to let a user to participate in a particular discourse of electronics startups and technological novelty.

Like the Processing programming environment from which it arose, Arduino also becomes a sandbox that creators tend not to leave once they've entered it. The sand can't easily be taken outside into the yard, partly because the tools themselves do not scaffold users to develop more sophisticated expertise with electronics, and partly because the Arduino community itself turns out to be a satisfactory, comfortable universe in which to conduct one's making.

On the one hand, we might praise devices like Twine and Arduino for creating their own ecosystems. But on the other hand, the development of those very ecosystems was always predicated on the promise of "graduating" into other communities—home DIY, or electronics manufacture, or even product development. The sandbox conflict turns these activities into private, rhetorical experiences with very little interest in the world outside the domain of their creation and support.

It's revealing to compare these computational examples of maker culture to other creative communities like knitting or woodworking. In such cases, we tend to find specialty infrastructures that facilitate and support them, like hardware stores and craft stores. They come with a clear understanding that the craft practice operates on the register of avocation and hobby *as well as* on the register of usable output. Here, the sandbox is viewed as a transitional object. It involves working with the actual materials that the broader world uses to accomplish goals, ratcheting up abilities with such materials, and eventually graduating into different levels of skill and professionalism. Of course, one can also knit or woodwork "for its own sake"—not all yarn must be rendered into sweaters, and knitters partake of their own communities for the sake of socialization as much as collaboration and support [7]. But the activity of knitting and woodworking is not *so completely* bound up with the latter type of activity that it entirely replaces the former.

Our Tiny Tinkering Platform

We've not just been thinking about the sandbox problem, but also we've tried to apply the hardware store metaphor to electronics tinkering platforms themselves, developing a small, inexpensive, and ubiquitous prototyping platform that can handle low-level varieties of input and output—creating basic sensing nodes. This "tiny tinkering platform" has a hypothetical target price of $1 per device (a constraint meant to help us avoid the expensive Twine scenario). We are focused on very low-cost, low-complexity hardware components, which have resulted in working prototypes that do things akin to the Twine and other devices but with far less capability out of the box. This approach to these sorts of systems assumes that a minimal, flexible, and very inexpensive platform could overcome obstacles around the price that broad experimentation requires, and might make permanent use in application-specific objects more viable.

In the process, we have been attempting to make design choices that would make our platform behave more like how we imagine a Twine would have to in order to acknowledge both the sand and the lawn sides of its life cycle—that is, not just to be talked about, but also to be used. Not just as a celebratory, "disruptive" lever in discourses about technology practice, but to be used so that it disappears into the background as infrastructure. While our project is a hybrid as well, we feel that tinkering with a platform should be straightforward, and solutions in the world should be simple and semi-permanent. Although we have experience in producing electronics as hobbyists and tinkerers, our inexperience with the architecture we have chosen has helped us to emphasize the difference between taking satisfaction in the creative practice itself (the sand) and taking satisfaction in the end product (the lawn).

MAKING IN PRACTICE

In order to get a sense of how our process has played out, we offer an overview of approximately two years of development of this simple remote sensing platform.

We started using the PIC10F200, a 6-pin microcontroller. In order to get around some of the missing functionality in this baseline part's lack of analog input, we produced a design using a potentiometer attached to its input pin to "tune" the attached analog sensor by adjusting its voltage. This let actuators to trip based on context. This worked but was not reliable across multiple sensors. We moved on to the PIC10F220, which added true analog input. It solved our sensor problem, but still didn't allow us any communication beyond immediately detectable output such as piezoelectric buzzers or LEDs. We did manage to revise the board and reduce the battery size while adding two analog inputs.

In order to try to leverage communications standards, we began to test the low-power PIC12F1552 microcontroller, which includes an SPI interface for peripherals like radio transceivers. As it was newly-released, we needed an updated PICkit programmer to be able to upload software to the chip—a process that, like all of the steps described here, involved ordering and waiting for parts to arrive. Programmer in hand, we implemented SPI for our radio. Although the datasheet for the microcontroller includes enough pins, we didn't have enough to control SPI due to an unforeseen issue: one of the 12LF1552's pins is unable to be addressed in code. It is exclusively a digital reset pin for embedded systems applications. As a result, SPI communication using this device was impossible.

We then shifted to the PIC12LF1840T48A, a version that includes an on-board radio transmitter. Its radio operates at 433 MHz, so we had to build custom hardware to receive signals at that frequency. Because this design is TSSOP (thin-shrink small outline package) rather than DIP, we also needed an adapter to use the chip on a breadboard. This also meant we had to test the radio transceiver we hoped to use with a new base-station design we are working on for this platform. That required the design, writing and testing of a communications protocol compatible with the transmission methods of the P12LF1840T48A. We also needed to build an antenna powerful enough for the transmitter to reach transmission ranges. All of this is ongoing, with setbacks, and while we're sometimes confused and frustrated, we feel as if we're building something using the native materials in which it needs to be made.

This kind of frustrating, sloppy, contingent, iterative production is what the knitting or woodworking equivalent of hardware making looks like. Being embedded in these lower-level electronics making practices reveals that constructing a prototype using an Arduino or its ilk are not really comparable to knitting, woodworking and the like. In most cases, the maker takes a pre-constructed prototyping board, a pack of sensors and uses open-source software libraries to assemble a project. It is a bit like buying a sweater, cutting off the tags, and calling yourself a knitter.

In some ways, the attributes that make platforms like the Arduino so popular—namely frictionless prototyping, relatively low cost of entry, and the availability of nearly ubiquitous sample code—are what prevent the novice from taking an uncomfortable step and moving into command-line code compiling C into machine instructions to be loaded into a microcontroller via a hand-built serial port. Toy projects, novelties, and small personal solutions can be cobbled together quickly and easily. At larger scale, however, the platform becomes wasteful and expensive. Indeed, one of the distinguishing features of the Arduino landscape is that there is no such thing as mass-produced Arduino project: with the exception of the fabricators producing the prototyping boards themselves, Arduino projects are one-offs, art projects, midterms, and hobbies; at their closest, they represent the output of a fledgling group of hobbyists who produce hardware for other hardware hobbyists. As a platform, the Arduino has dramatically increased the population of people interested in producing, sharing and being excited about electronic hardware manufacturing. At the same time, the dearth of "exit strategies" and successes for the platform may mean that Arduino for most people is both the beginning and the end of a hobby creating electronic objects.

THREE ENDGAMES FOR MAKER CULTURE

Given the implications of this sandbox problem in today's maker culture, we observe three possible endgames for it. We phrase these in terms of three configurations of the sandbox conflict—new rhetorical framings to think about the role of making in broader practice.

The first configuration involves no change to the status quo. Those inside the sandboxes continue servicing their own desires, in some cases producing industries around themselves, but not affecting the outside reference activities to which the maker activity aspired or laid claim. Makers remain makers, but nothing more. In this mode, making remains a self-serving meta-hobby that *refers* to a real-world activity but does not interact with it. While this outcome may seem like a failure of the revolutionary rhetoric surrounding making, it is really an honest appraisal of the outcomes of the current modes of production within making culture: making is a creative and expressive practice, but a practice unto itself.

The second outcome denies that the sandbox exists at all. Prior reference activities are abandoned in favor of the sandboxed "maker" activity. Here we find a growing commercial ecosystem to support joining and becoming a part of this "movement," supported by specialist shops like Adafruit Industries, Sparkfun, and Seeed Studios—not to mention "reference" hardware manufacturers like Arduino. Even industrial behemoths are getting into the game, as Intel has released enthusiast-oriented hardware like the Galileo both to position themselves as responsive to the maker community and to stake a claim within it.

Finally, the third end game situates the sandbox as a playground for evolutionary effort that "graduates" to professional practice. Those inside maker activities should recognize that the being a maker risks self-reference unless the sandbox itself is abandoned once it offers enough lessons to inform a broader set of activities. People risk aspiring to be "makers" instead of makers of *things.*

In practice, all of these configurations look very similar. In examining the goals of making, attending to both the sand and lawn is essential. By considering the future of making—supporting people who make these real, useful, things—the third option is the most appealing to us. It might also seem retrograde and even boring. Ultimately, making in this third style really just looks like ordinary craft or traditional engineering. This outcome may even be incompatible with the very concept of "maker culture." The point of a maker culture is to reject the desirability of a return to the reference domain in favor of the spawning of new sandboxes. Put differently, maker culture may be incompatible with anything but its own self-propagation.

ACKNOWLEDGEMENTS

This work was supported by the Intel Science and Technology Center for Social Computing (ISTC-Social). Thanks especially to the reviewers for their feedback.

REFERENCES

1. Ames, M. et al. Making Cultures. *Ext Abstracts of CHI'14*, (2014), 1087–1092.
2. Anderson, C. *Makers: the new industrial revolution*. Crown Business, New York, 2012.
3. Blikstein, P. and Krannich, D. The Maker Movement and FabLabs in Education. *Proc. IDC'13* (2013), 613–616.
4. Goodman, E. and Rosner, D. From garments to gardens. *Proc. CHI'11*, (2011), 2257–2266.
5. Hertz, G. *Critical Making*. Telharmonium Press, 2012.
6. Krannich, D. et al. Digital Fabrication for Educational Contexts. *Proc. IDC'12*, ACM (2012), 375–376.
7. Kuznetsov, S. and Paulos, E. Rise of the expert amateur. *Proc. NordiCHI'10*, (2010), 295–304.
8. Leduc-Mills, B. et al.. Evaluating Accessibility in Fabrication Tools for Children*. IDC'13*, (2013), 617–620.
9. Lindtner, S., Hertz, G., and Dourish, P. Emerging Sites of HCI Innovation. *Proc. CHI'14*, (2014), 439–448.
10. Lindtner, S. and Li, D. Created in China:. *interactions 19*, 6 (2012), 18–22.
11. Mellis, D. and Buechley, L. Collaboration in open-source hardware. *Proc. CSCW'12*, (2012), 1175–1178.
12. Mellis, D. and Buechley, L. Scaffolding creativity with open-source hardware. *Proc. C&C'11*, (2011), 373–374.
13. Mellis, D.A. and Buechley, L. Case studies in the personal fabrication of electronic products. *Proc. DIS'12*, (2012), 268–277.
14. Ratto, Matt. Critical Making. *The Information Society 27*, (2011), 252–260.
15. Rosner, D. and Bean, J. Learning from IKEA Hacking. *Proc. CHI'09*, (2009), 419–422.
16. Sadowski, J. and Manson, P. 3-D print your way to freedom and prosperity. http://alj.am/1kanblT.
17. Tanenbaum, J. et al. Democratizing Technology. *Proc. CHI'13*, (2013), 2603–2612.
18. Tanenbaum, K. et.al. Critical Making Hackathon. *Ext. Abstracts CHI'14*, (2014), 17–20.
19. About us | Raspberry Pi. http://raspberrypi.org/about.
20. Supermechanical : Twine. http://supermechanical.com/