Cardboard Machine Kit: Modules for the Rapid Prototyping of Rapid Prototyping Machines

Nadya Peek
MIT Center for Bits and Atoms
peek@mit.edu

James Coleman
A. Zahner Co.
jcoleman@azahner.com

Ilan Moyer
Shaper Tools Inc.
ilan@shapertools.com

Neil Gershenfeld
MIT Center for Bits and Atoms
ng@cba.mit.edu

ABSTRACT
Digital fabrication machines (such as laser cutters or 3D printers) can be instructed to produce any part geometry within their application space. However, machines' application spaces are not easily modified or extended. How can we enable the production of application-specific computer-controlled machines by machine building novices? How can we facilitate rapid prototyping of rapid prototyping tools? We propose a novel set of modules, the Cardboard Machine Kit, for the construction of digital fabrication machines. These open-source modules are implemented using cardboard frames, stepper motors, and networked electronics controlled through a Python library. We evaluated the kit both through machine building workshops and by studying the usage of the kit in the wild. In the wild we observed more than 500 novice machine builders who built 125 different machines for 15 different application types. We argue that this breadth demonstrates the efficacy of this modular approach. Finally we discuss the limitations of the Cardboard Machine Kit and discuss how it could inform future machine building infrastructure.

ACM Classification Keywords
J.6 Computer-aided manufacturing (CAM): Computer-Aided Engineering; H.5.2 Prototyping: User Interfaces; H.5.2 User-centered Design: User Interfaces

Author Keywords
Digital fabrication; Machine Building; CNC; CAD/CAM; Cardboard; Prototyping

INTRODUCTION
Digital fabrication machines are taking a central place in HCI research and discourse on making [11]. Digital fabrication machines intended for personal use such as desktop 3D printers, sub-100W laser cutters, or desktop CNC milling machines are becoming more accessible through decreasing cost and increasing usability. The machines allow unlimited variation on part geometry within their application space; as long as it

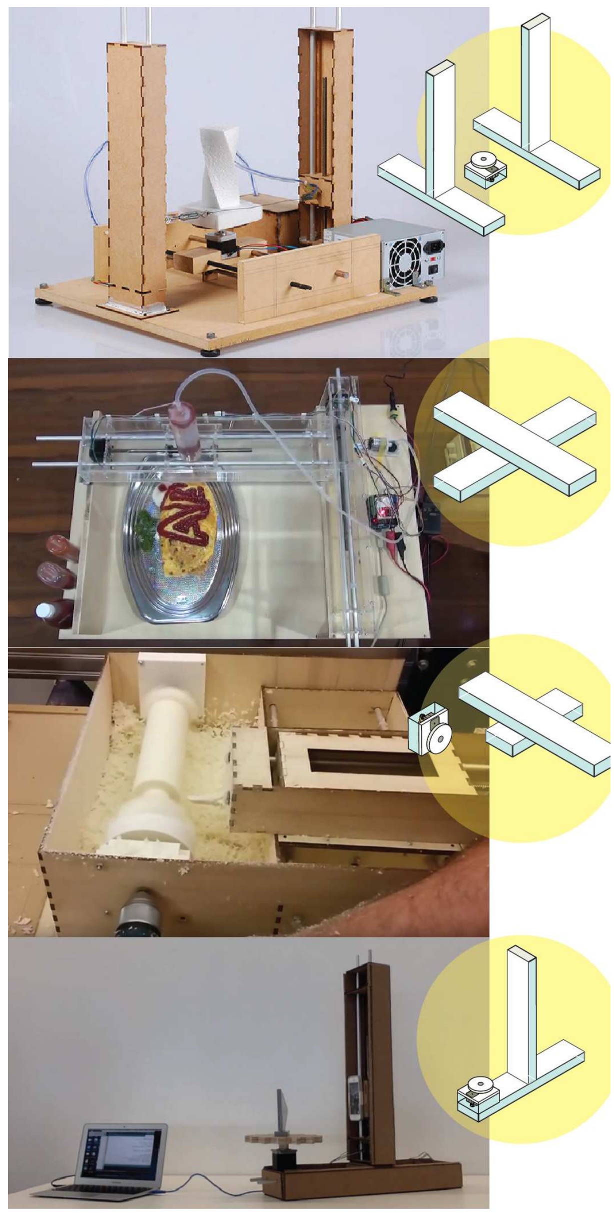

Figure 1. Machines built in the wild using the Cardboard Machine Kit: A 4-axis hot wire cutter by Fablab Monterey, an omelette ketchupping machine by Fablab Kitakagaya, a lathe by OpenDot, and a 3D scanner by Fablab Pueblo. Schematics of motion modules shown on the right.

CHI 2017, May 6-11, 2017, Denver, CO, USA.
ACM ISBN 978-1-4503-4655-9/17/05.
http://dx.doi.org/10.1145/3025453.3025491

fits within the work envelope, any line can be cut on on a laser cutter or any 3D geometry printed on a 3D printer.

However, these machines are not easily modified. It is non-trivial to extend their work envelopes or change their end effectors. A machine often needs to be replaced entirely to extend its work area. Robotic arms have interchangeable heads, but have limited work envelopes (especially with respect to their size), comparatively low precision, complicated programming workflows. Adding closed-loop control to a machine or another axis to the motion platform entails remaking their control electronics.

The motion systems of different digital fabrication machines are similar. End effectors need to move in XYZ-space, and commonly do that with 3, 4, or 5-axis control. Depending on the application of the machine, the hardware can be optimised for speed, stiffness, and size. For example, a laser cutter needs to have optics that move fast, but they encounter no cutting force. In contrast, a milling machine used for titanium does not need to move quickly, but needs to be able to apply a lot of force on the workpiece. Different motors and drive trains might be chosen accordingly. Binding a machine to a specific application space makes it easier for the machine designer to optimise the machine for that work flow. These optimisations are important design decisions for work horse machines that are expected to be producing parts in high volume.

However, personal digital fabrication does not need high volume production. Personal fabrication is instead focused on providing high precision tools to a very diverse set of users. These diverse users want to produce a correspondingly diverse set of products. The choices traditionally made in digital fabrication machine design are based on assumptions of high volume production by expert users. The resulting machines not well suited for possible diversity of personal fabrication. They are not easily extensible or modifiable. How can we avoid the limitations of past designs in future machines? How can we create machines that represent the diversity of their users? Can machine design be done by people who are not experts in machine design, but are experts in their application space (the machines' users themselves)?

We present modular machine building infrastructure in the form of the Cardboard Machine Kit. This infrastructure is meant to enable the production of digital fabrication machines that mirror the diversity of the goods produced on digital fabrication machines.

The Cardboard Machine Kit contains a novel set of primitive modules for machine construction. A schematic representation of primitive motion modules is depicted in Figure 2. This set includes open-source designs for stackable single degree-of-freedom motion platforms (both linear and rotary), novel networked control electronics, and novel software for interfacing with the machines. The modularity of these components both in the motion platform and the control system enables users to design machines in a way that parallels current software design practices. Harnessing modularity in machine design is one of the key insights of this paper.

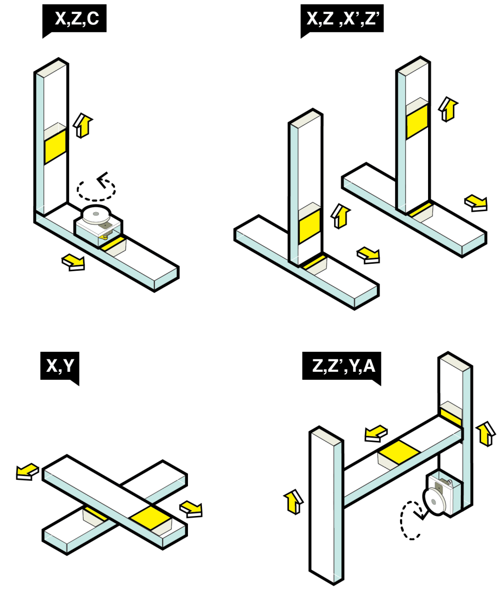

Figure 2. Configurations of primitive modules into machines. Clockwise from the top left, we have a machine with three degrees of freedom (two linear and one rotary), four degrees of freedom, two degrees of freedom, and three degrees of freedom with one redundant axis.

We conducted a series of workshops with university students, researchers, and servicepeople using the Cardboard Machine Kit. During these workshops we evaluated both the efficacy of its design and of the design of its documentation, updating both accordingly. We then conducted a distributed build cycle, where participants in fab labs and makerspaces built machines using the Cardboard Machine Kit. Some machines are shown in Figure 1 and more are shown in the section 'In the Wild'.

During the workshops we found that the Cardboard Machine Kit enabled people with a maker background to build novel machines for digital fabrication in a few days. Beyond our expectations, we also found that people with no background in making or engineering were able to use the kit to produce machines along a similar timeline. Our original expectation was that a modular approach to machine building would allow users with some familiarity (such as engineering undergraduates) to make machines. However, we observed a much more diverse and less expert group of users make machines from the kit in the wild.

This work contributes a novel technical system for machine building. This system employs modularity in motion design and networks in control system design which are key insights that together form a novel approach to machine building. We describe the workflow for the system in a walk through and describe the system's implementation. Observing the system's use in workshops and in the wild, we learn more about the benefits and limitations of a modular machine building system.

Finally we discuss how the insights from deploying the kit can inform future work in machine design.

RELATED WORK

Digital fabrication and maker culture have been increasingly popular topics of study in the past decade. New materials for digital fabrication and methods for digital fabrication are regularly published [9, 25, 6, 21]. HCI research has employed digital fabrication for making specific kinds of products [24, 14, 20, 5, 7]. How DIY and maker culture is informed by and informs 'professional' technology is studied in detail [3, 1]. The research presented here draws from these findings, using insights from research on maker culture to imagine alternative and inclusive infrastructures for digital fabrication.

The research presented here draws from the research done on early internet infrastructure, especially *End-to-end Arguments in System Design* [19], which argues for applications to be implemented at the end points of networks.

Little Bits present a modular approach for hardware construction [2] similar to the *Lectron Set* Braun released as an educational tool in the 1970s. We similarly use a modular construction-set approach, but for machine design instead of electronics.

Mellis et al. use the term 'untoolkit' to refer to an extensible framework for prototyping electronics [12]. Their approach to accessibility and breaking through the confines of the 'kit' has influenced this work. Jacobs and Zoran study the use of digital fabrication tools in diverse populations [8]; we draw from their insights into accessibility of digital fabrication as well.

The unconventional digital fabrication tools FreeD and Shaper Tools' Origin allow a digital model and its physical prototype to constantly reconfigure each other [26, 16]. The implications for bidirectional design integration are also explored by ReForm [23]. We consider these unconventional workflows while drawing up the required capabilities of our machine building infrastructure.

CARDBOARD MACHINE KIT

We present a set of primitive modules for machine building. All of the designs in this section including mechanical designs, fabrication settings, board design, firmware, and software are open-source and freely available at http://mtm.cba.mit.edu. An excerpt of their documentation is shown in Figure 3.

Walk through

A machine builder seeking to prototype a machine first may consider the selection of the end effector to be employed. We are not including the design of end effectors in this paper, although it is an active research topic [22]. For example, the machine builder may want a kind of pen holder to make a drawing machine.

Next the machine builder might consider the work envelope required for their machine. In the case of a drawing machine, perhaps 30×30 cm is a good XY size, and for pen up and down 10 cm is sufficient. Then the user needs to determine how to use the primitive motion modules to create the motion

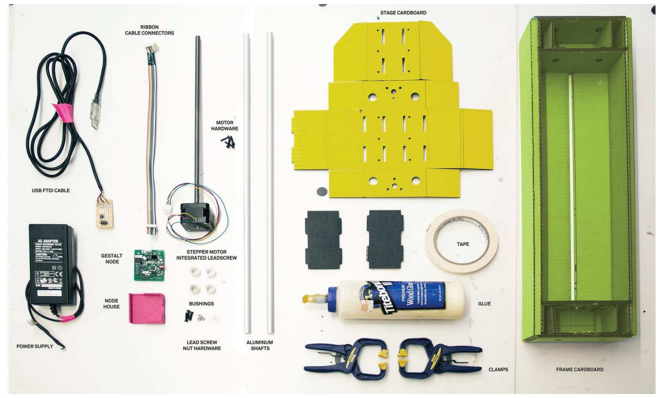

Figure 3. An excerpt of the Cardboard Machine Kit step-by-step assembly instructions, as linked to from http://mtm.cba.mit.edu.

required. Some instantiations of motion using primitive modules are shown in Figure 2. Let's imagine the machine builder selects two redundant Y modules, a bridging X module, and finally a Z module to bridge both. For reference, this same selection was made by the designers of the machine in Figure 7.

By default, the linear motion module has 30 cm of travel. To customise the length of the module, the machine builder specifies a different length in the parametric design file that generates the module's laser cutting files[1]. There the machine builder can also customise the material thickness (by default this the thickness of the commonly available tri-fold presentation board cardboard), and the module's width. Example output for sending to the laser cutter is shown in Figure 4.

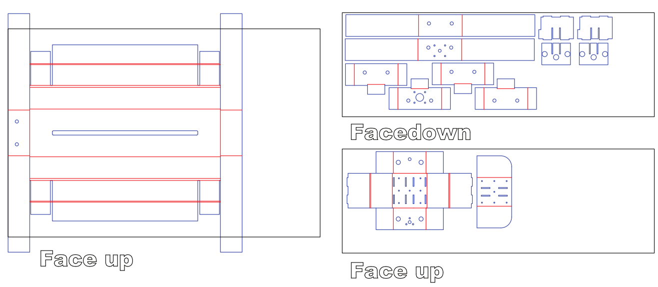

Figure 4. Laser cutting files for the cardboard stages as output from the parametric design file, where red is set to score and blue is set to cut through. The face up and face down is relevant in case the cardboard has a 'good' side.

The machine builder then laser cuts the designs from the material they have selected. We encourage the use of cardboard in the documentation, as it is inexpensive, robust, and easy to modify in case of initial design flaws. However, the design also accommodates other more permanent materials. The laser cut material is folded and glued into the frame and stage of the motion module. Additional hardware such as the motor, the guide shafts, and the bearings are assembled at this point.

[1]The size of the resulting file might be larger than the laser cutter the machine builder has access to. For this case, we have included designs for dovetailing parts together.

After building the individual stages, they are assembled into the desired machine configuration. This can be done using the pre-made press-fit connection points, by building custom connectors, or by simply gluing the cardboard stages together.

Then the machine designer can connect the kit's networked control node circuit boards to each motor. The nodes are then wired together and to the machine builders's computer. Extra nodes can be added to the network at any time, should the machine builder find the need for another axis, sensor, or control point. A overview of such a network using the boards we implemented is shown in Figure 5. This network of control nodes can be addressed from a Python software library on the machine builder's computer. Several default kinematic configurations are included with the library, or the machine builder can specify their own configuration. In this case, the machine builder needs to specify [X, Y1, Y2, Z] nodes in a direct drive configuration to accommodate their redundant Y axis.

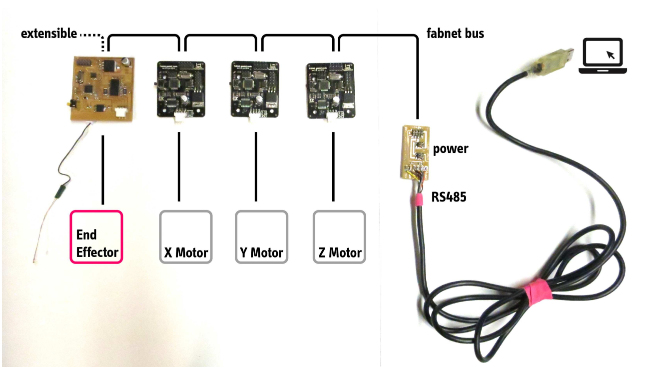

Figure 5. A network of PyGestalt control nodes that make up the electrical controls for a machine (in this case, a 3-axis 3D printer).

To configure the software control with the individual nodes, the control library prompts the machine builder to identify each control node (e.g. Y1) they have specified by pressing its included button in order. A light flashes on the control board to acknowledge that an ID has been assigned. Once the library is configured to communicate with the machine, the machine builder can send the machine moves such as [0,0,1] (pen up) or [24, 12, 0] (move to (24, 12)) or [0,0,0] (move home). The machine builder can send other commands as well, such as speed changes, or an instruction to change the current limit setting.

The machine building exercise in this walk through can easily be accomplished in a few hours. If the machine builder needs different application-specific items such as a complex end effector, a new type of control board (e.g. for adding a sensor), custom kinematics (e.g. for a delta bot), or custom actions (e.g. for reading the added sensor), then this will extend the amount of time required. These parts can be developed in parallel with the first machine prototype though, allowing for concurrent testing of subsystems as is the norm in software development.

Implementation Details

The Cardboard Machine kit comprises software, firmware, board designs, mechanical designs, and a bill of materials that we will provide more detail on in this section. The decisions made for this implementation were informed by some overarching design principles. We prefer modularity over monolithic implementations, because we believe it enables reusability and prevents system-wide failures. We prefer end-to-end design principles, because we believe efforts made in infrastructural developments can and should be enjoyed by diverse applications on that infrastructure. Finally, we believe in open design and reusability, because that makes it easier for the next machine builder to develop their custom version. In service of the last design principle, we provided all designs with open licenses and by making a concerted effort to ensure the materials required were readily available to others, for example by choosing common components that are available off-the-shelf.

Table 1 has contains the bill of materials for a single stage, including the control electronics.

Item	**qty**	**cost (USD)**
Tri-fold Poster Cardboard	1	4.16
Nylon bearing, flanged, 3/8" ID 1/2" OD	4	2.93
Aluminium tube, 3/8" OD	4’	9.26
Stepper motor N17 w/30cm lead-screw and nut	1	27.45
M3 fasteners 12mm	3	.30
M3 fasteners 8mm	4	.40
M3 locknuts	3	.30
PyGestalt stepper motor control board	1	15.20
total		**60.00**

Table 1. Bill of materials for a cardboard linear stage module.

The frames for the motion modules are made out of cardboard in our base design. The cut files for the cardboard can be generated using a parametric design file we made in Grasshopper and Rhino. The machine builder specifies the dimensions of the material and the dimensions of the desired motion module, and the Grasshopper program generates a cut file accordingly. An example of a cut file is shown in Figure 4. Our reference motion stages are specifically made with Pacon brand tri-fold presentation cardboard, which comes in many different colours. We chose to use cardboard in our base design because it is the ultimate mutable material for prototyping—it is very easy to glue onto, to cut and change, or to recycle. Cardboard is a democratic material; its wide availability translates to wide participation. Cardboard is not as stiff or strong as e.g. metal, but it is much cheaper and lighter. Laminating several layers of cardboard such as in our design improves the stiffness, making it comparable to a frame of made of T-slot assembled acrylic or plywood of 1/8" thick such as is used in the Fab@Home [10]. Aluminium extrusion such as the 80/20 framing system would be at least ten times higher in cost and would require machine builders to use tools such as chop saws or drill presses to get material to size. We would argue that those kinds of tools are more difficult and dangerous for a user to use, and that user error has much more serious consequences for the resulting machine parts. Using a laser cutter outsources the required precision in cuts to the machine instead of the user, making it easier to reliably produce useful parts.

The cardboard motion platform on the linear stage modules has four nubs in the style of Legos on the moving platform. These nubs match to hole patterns on the edges and middle of the frame. This way, one stage can easily be attached to the moving platform of another stage. This is to facilitate assembling the stages into different machine configurations. However, machine builders can also assemble the stages using many other methods such as glue, fasteners, or by replacing the moving platforms in the stages with parts that integrate into the next stage.

The drive train is a 4-start 2mm pitch NEMA 17 stepper motor with aluminium guide rails sitting on nylon bushings. The 4-start lead screw enables fast travel (8mm/revolution) while still using the force of a lead screw. We had a run of custom motors with matching wear-compensating lead screw nuts made for the workshops, but identical motors are available off the shelf. The standard design accommodates 30cm of travel for the stage, but the parametric design can be modified to accommodate an arbitrary length of leadscrew. The motor and leadscrew are clearly overkill for the frame (the motor has enough force to easily rip the cardboard frame apart), but were chosen to minimise number of parts on the bill of materials, as well as to provide reusable hardware for future machine design iterations.

The networked control nodes use the fabnet extended RS-485 protocol for communication as described in [13]. The stepper motor boards we designed for the Cardboard Machine Kit contain an AVR Atmega328 microcontroller and an an Allegro A4983 stepper motor driver for receiving packets and controlling the motors. They furthermore contain indicator lights, a potentiometer for setting the current limiting to the motors, and a differential bus transceiver chip for communication (a 75176AD). The firmware is written in C. The Cardboard Machine Kit stepper motor boards are the black boards shown in Figure 5. Their designs and firmware are available at http://github.com/imoyer/086-005.

For controlling the machines, we present the PyGestalt library for machine control. PyGestalt is open source and available online at http://github.com/nadya/pygestalt. PyGestalt allows us to create *virtual machines* for controlling the physical machines with. For example, if we have an XY stage that moves with one motor controlling the X and a separate motor controlling the Y, the virtual machine will assign X motion packets to the X controller node and Y motion packets to the Y controller node. If we have an XY stage that moves with some form of parallel kinematics (where both motors are used in both X and Y positioning), then the appropriate components of the moves will be calculated by the virtual machine and assigned to the A node and B node. Conveniently this means that if we create an application, such as a program that generates coordinates to send to a machine, we can still apply that to several different machines as long as we swap out the virtual machine controller. The mechanical implementation details are separated from the user interface.

Example code for setting up the `virtualMachine` using H-bot kinematics with MXL pulleys for XY and a leadscrew with 8mm of travel per rotation for Z below:

```
def initKinematics(self):
        # drive components of h-bot.
        # Inputs are A/B stepper motors,
        # outputs are X/Y in machine coordinates.
            self.aMotor = elements.elementChain.forward(
                [elements.microstep.forward(4),
                elements.stepper.forward(1.8),
                elements.pulley.forward(2.03),
                elements.invert.forward(False)])

            self.bMotor = elements.elementChain.forward(
                [elements.microstep.forward(4),
                elements.stepper.forward(1.8),
                elements.pulley.forward(2.03),
                elements.invert.forward(False)])

        self.zAxis = elements.elementChain.forward(
                [elements.microstep.forward(4),
                elements.stepper.forward(1.8),
                elements.leadscrew.forward(8),
                elements.invert.forward(True)])

        xyKinematics = kinematics.hbot()
        zKinematics = kinematics.direct(1)
        compoundKinematics = kinematics.compound(
                                [xyKinematics, zKinematics])
        self.stageKinematics = compoundKinematics
```

Using this `virtualMachine` object the user can send XYZ coordinates in millimeters without being concerned with the particular implementation of the physical machine (which pulleys, what lead screw pitch, etc.) or the machine needing to move both the A and B motors to move in X or Y. This introduces a layer of modularity that enables the reuse of toolpaths in machines with different machine implementations. A very simple example script for moving a machine able around is shown below:

```
import virtualMachine

if __name__ == '__main__':
        # The persistence file retains which node has been
        # assigned which ID
        myMachine = virtualMachine(persistenceFile="mm.vmp")

        # This is for how fast we will move in mm/s
        myMachine.abzNode.setVelocityRequest(6)

        # Some random moves to test with
        moves = [[15,0,0],[15,15,0],[15,15,15],
                [0,15,15],[0,0,15],[0,0,0]]

        # Move!
        for move in moves:
                myMachine.move(move)
```

The `virtualMachine` could be the one implemented above, or one with completely different paramaters. The `virtualMachine` furthermore has a set of standard functions implemented such as `move` or `setCurrent`. More functions can be added by the user as required.

The Cardboard Machine kit comprises software, firmware, board designs, mechanical designs, and a bill of materials that were described in this section. The approach of this implementation is to create base infrastructure for machine building that can be augmented by the machine builders themselves.

Limitations

There are clear mechanical drawbacks to using modular stages that couple and decouple as machine tools. Such an assembly of parts will never be as stiff as a custom-built monolithic machine, even if they are made of materials that are more stiff than cardboard. Stiffness is required in machine tools to withstand cutting force with minimal deflection and vibration at the tool tip. Connections between parts might be particularly susceptible to vibration or introduce backlash. Using single-degree-of freedom modules furthermore implies the use of serial kinematics, meaning that stages are stacked on top of each other. This means that the X axis needs to carry the weight of the Y axis on top of whatever machining forces the machine might encounter. This has implications for the rate at which the machine can accelerate and also how slop in the system might compound.

There are also limitations to using a networked approach for the control system. The bandwidth of the network becomes the limiting factor for communication with the machine. In our case, this is limited by RS-485, which within the machine communicates at 35Mbit/s. If acceleration and deceleration curves need to be used at high resolution, this could become a problem. For small-format milling, laser cutting, or 3D printing though, 35Mbit/s is sufficient bandwidth to have comparable performance to conventional machine tools.

By making machines out of modular mechanical and control parts, we introduce more versatility through reconfiguration, more robustness by making broken parts easy to swap out, and lower cost by reducing the design cost of each individual machine. These benefits come at the expense of not achieving the same optimal designs as purpose-built machines.

A similar discussion was held during during the introduction of packet switching networks, which served as the foundation of later protocols like TCP/IP. Some people argued that packet switching was a less optimal way to transmit data than dedicating entire connections between nodes to individual communications, as in the telephone networks of the time [17]. Later, TCP/IP was developed to create an "effective technique for multiplexed utilization of existing interconnected networks" [4]. TCP/IP, built on top of packet switching networks, enabled distributed heterogeneous networking at enormous scale. A fundamental technology change gave rise to the proliferation of many different networks across the globe.

We hope that modular infrastructure for machine building, of which we present a very simple example, will grow in its robustness and sophistication as it is demonstrated to be hold value over conventional machine building approaches.

WORKSHOPS

Using the Cardboard Machine Kit described in the previous section, we ran four formal workshops. Workshops 1 and 2 were held in a university with a mix of undergraduate and graduate students of all majors (60 students per workshop).

Figure 6. Workshop participants building machine parts.

During those workshops, students came to a one-hour lecture, one-hour of lab demonstration, and had eight hours of lab access to work on their machines. Some images from this workshop are shown in Figure 6. Workshop 3 was held at a computer graphics conference in one four-hour slot for 30 participants [15]. For Workshop 3 we pre-cut cardboard to assemble linear stages of a default length due to both time limitations and lack of laser cutter. Finally, we taught in Workshop 4 a series of four one and a half day workshops on a military base with servicepeople (20 students).

For all four workshops, students were provided with the full bill of materials listed in Table 1. For Workshops 1, 2, and 4, the students additionally had access to a full digital fabrication facility.

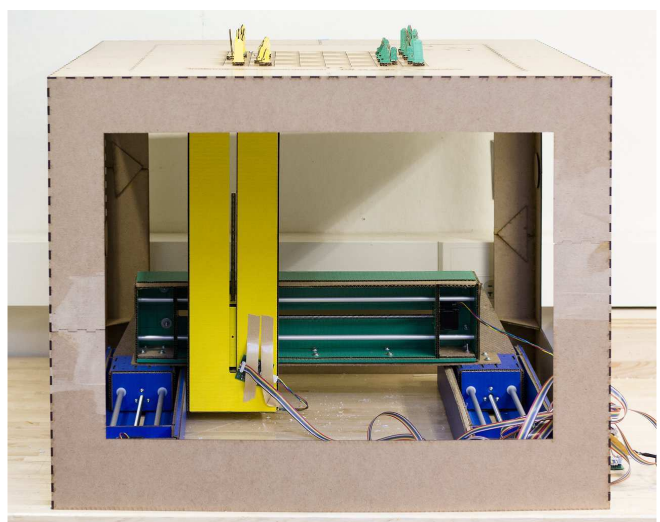

Figure 7. A real mechanical Turk–a chess playing table. For this machine, the designers interfaced the machine with an existing chess-playing library, so that a player could play chess against a computer in physical space.

In Workshops 1 and 2, the participants collaborated in teams of 10-15. Figure 7 shows a chess playing machine made in the first workshop. The chess machine was one of four machines produced. It includes a chessboard and app for playing chess on the table remotely. The end effector in this case is a magnet which can move the pieces around on the table. To achieve this, the students split into groups concerned with app development, UI, table design, controls, chess piece design, and mechanical implementation. The end effector development quickly started working on the possibilities of magnetic interference between pieces, especially when moving the knight. Despite the final motion system not being ready, the sub team was able to test with a minimally viable prototype of the machine. Being able to use a minimally viable prototype like this greatly contributed to the subteams getting everything working together, despite leaving only a few hours for system integration before their final presentation.

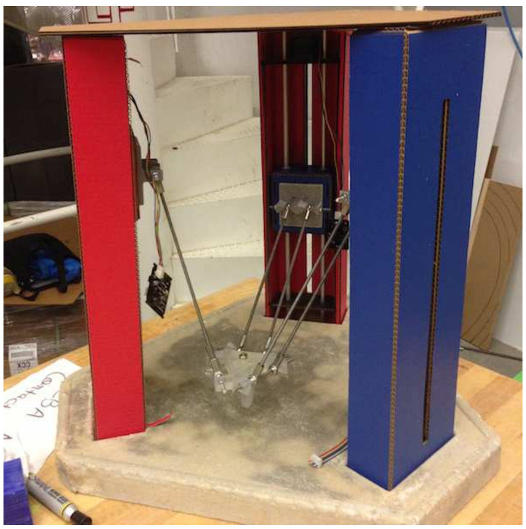

Figure 8. A machine built that uses delta bot kinematics for moving around a tongue-shaped end-effector for "subtractively decorating cakes".

Figure 9. A 3-axis hot wire cutter for cutting foam. Many hot wire cutters were produced, including 4- and 5-axis ones such as the first machine in Figure 1. Hot wire cutters are a very easy way to rapidly prototype complex shapes out of foam, which perhaps explains their popularity.

Figure 8 shows a artificial tongue controlled by a machine built delta bot kinematics built by a group participating in Workshop 2. The purpose of this machine was "subtractively decorating cakes". Another machine built in the same workshop decorated cakes by extruding frosting, which was called "additively decorating cakes". To be able to make this subtractive decoration machine, the students needed to augment the existing machine control library with delta bot kinematics, as well as design custom delta bot hardware. Despite the relative sophistication of their software programming, they did not see any need to replace the simple cardboard implementation of the modules. This shows how the kit supports system integration between complex and simple components.

A hot wire cutter made by the computer graphics researchers who participated in Workshop 3 is shown in Figure 9. As with the example machine from Workshop 2, the participants were skilled in software engineering but not in fabrication. Their machine due to being constructed with only three degrees of freedom had limitations for what kind of geometries could be cut. The participants wrote a software module to create designs that fit the machine's geometry constraints within the confines of the three hour workshop.

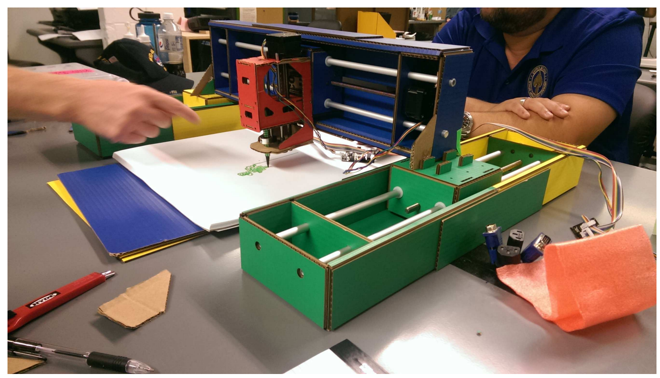

Figure 10. This is a pen plotter that uses two stages for the Y-axis. That redundancy eliminates the need to cantilever the X-axis. The 2 stages for the Y can be tied together in the VM, so they always move in unison. This machine was one of four constructed in Workshop 4.

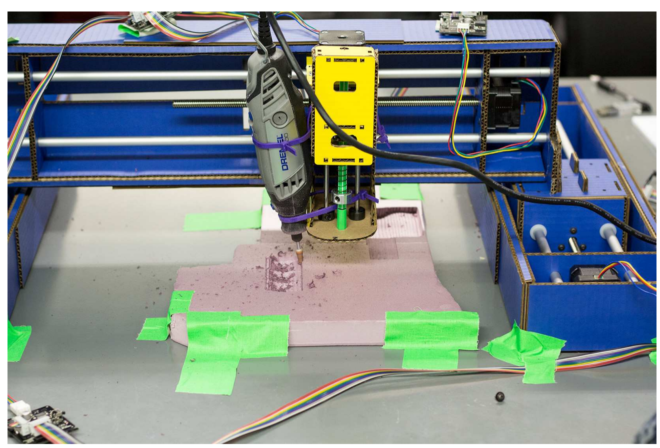

Figure 11. In the spirit of rapid prototyping of rapid prototyping machines, these users first built a pen plotter such as the one in Figure 10, but then quickly decided they actually wanted to build a foam milling machine, and make a few quick modifications accordingly.

Figures 10 and 11 are from the final workshop which was conducted with active military servicepeople. This workshop was part of a four part introduction to digital fabrication, where the first two classes taught the use of CAD and laser cutters. Many of the participants did not have college degrees. The workshop was 9 hours long, split over two days. For this workshop, the participants first constructed a pen plotter with the modules. After the reference machine, the participants could customise or redesign the machine.

Figure 11 shows the machine modification that was made by one of the groups in Workshop 4. Despite their limited fabrication skills and the time limitation, they decided to turn their pen plotter machine into a milling machine. After attaching the Dremel as a milling spindle, they observed that the machine was now subjected to higher force as the Dremel encountered foam. Whereas with the pen they had only attached the X stage with a single cardboard tab, they now modified the structure of the machine to give better load transfer and stability. This kind of modification demonstrates experiential learning by the workshop participants. The participants of Workshop 4 self-reported that the machine building exercise was “a life-changing experience". Despite their occupations as tradespeople who service massive ships, vehicles, and other military infrastructure, they reported that based on their experience building machines from scratch, they now had “new superpowers, or the ability to understand and make any kind of machine".

IN THE WILD

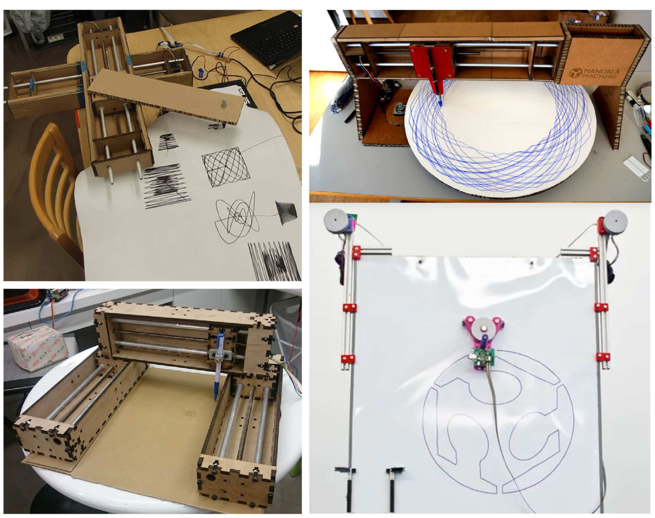

Figure 12. Different types of drawing machines produced. Clockwise from top left, there is one made with two modules at Fablab Montreal; a Mandala Machine one made with one linear and one rotary module at Fablab Reykjavik, one made with a pen hanging from pulleys at Fablab HRW, and one made with three modules made of plywood at Fablab Tainan.

The Cardboard Machine Kit was also made available to machine builders who did not participate in the formal workshops the authors taught in person above. The kit’s full bill of materials as described in Table 1 (with the exception of the cardboard) was sent to 64 separate maker spaces and fab labs throughout the world. More than 10 other maker spaces sourced parts for the Cardboard Machine Kit independently.

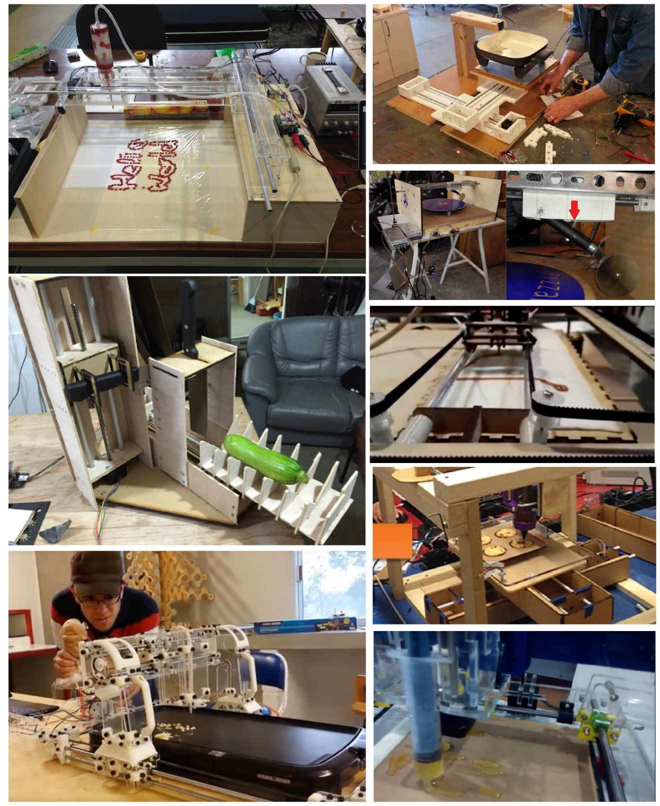

Figure 13. Many machines dealt with some aspect of food preparation. For example, clockwise from top left: a ketchup printer from Fablab Kitakagaya; a pancake printer from CITC Fablab Alaska; a pizza cutter from Fablab Barcelona; a vegetable peeler from Fablab Sendai; a cookie froster from AS220; a chocolate printer from Fablab Trivandrum; a vegetable cutter from Fablab Hamamatsu.

These machine builders were participating in a one semester distributed course on digital fabrication. As one of the two week assignments of the course, they were tasked with producing a machine. In the course they had previously learned about digital fabrication including laser cutting, electronics fabrication, programming, and 3D printing. However, there are no prerequisites for the course, and many participants were complete engineering novices. The participants had access to the online documentation of the kit as well as a video lecture by the authors. They furthermore had access to the tools of their respective maker spaces, including a local resident expert maker. The results presented are from two separate cohorts of the course.

In total, 125 machines were produced over a two week period. Figures 12, 13, 14, and 15 show some of the machines that were constructed. These machines included food preparation machines, music playing machines, and CNC hot wire cutters. The documentation pages the students are required to produce in the class were used to tally the machines into categories. These categories are shown in in Table 2.

Figure 14. Odd machines produced included a machine that injects coloured liquid into bubble wrap bubbles to make a kind of dot-matrix printer by Fablab Torino (top); sand gardening machines by Opendot Milan and the LCCC Fablab (middle); and music making machines by the Oulu and LCCC Fablabs (bottom).

Less than a quarter of the machines produced were made as general purpose digital fabrication machines such as a laser cutter, 3D printer, or mill. Many more application-specific machines were built, such as the tube cutting machine shown in Figure 15, or a machine for producing shredded daikon radish, or a machine for moving a laser pointer in a random fashion to keep a cat amused.

None of the participating students were unable to get something to work. However, not all the students were able to achieve all of the ambitions they had for their machine projects. For example, the tortilla maker shown in Figure 13 had a beautiful machine frame that involved significant design and fabrication work, but a control system didn't yet make the best looking tortillas. When we developed the Cardboard Machine Kit, we specifically had in mind the trouble which might arise when a user is unable to build a machine due to a lack of a particular subset of skills needed. The tortilla machine makers obviously have the CAD and fabrication skills required for machine building, but did need the electronics and software framework in the kit to get started on controlling their machine. The daikon radish peeling machine's builders went through several spirals of development for each separate subsystem: their first peeling end effector did not work, requiring design iterations. Their first user interface was simply a command line prompt, but later they built an app so they could control the machine from a smartphone. Their first mechanical design was cardboard, later replaced by a colourful 3D printed design. We observed clearly how the kit's modular components

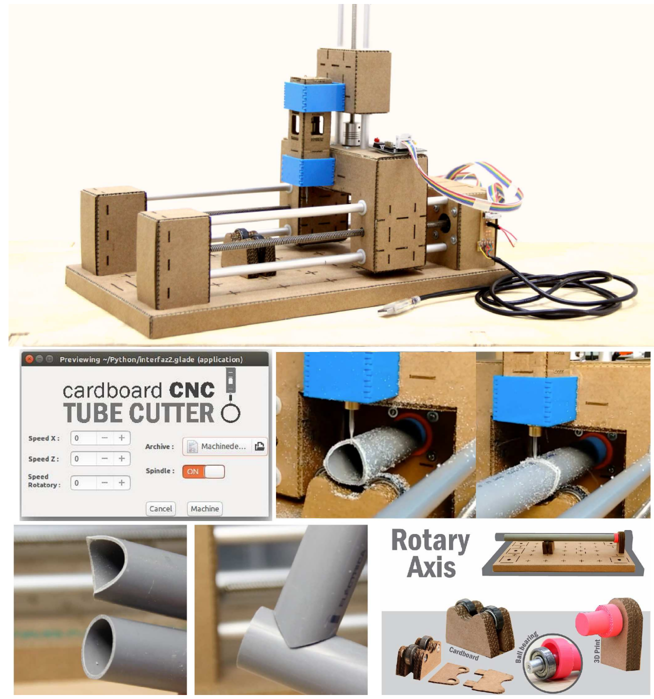

Figure 15. This application-specific milling machine is designed to cut PVC tubing for joining at different angles. It is a clear example of an application that requires computer control (for cutting precise curves) but does not require high throughput or need to withstand high cutting forces. Despite being made of cardboard, it is more than stiff enough to handle the cutting force from the spindle end effector. Made by FabLab Tecsup students Fabio Ibarra, Gabriela Mojoli, Jesús Valencia, Roosvelth Cántaro, and Jorge Valcárcel.

encouraged subsystem iteration and assist with later system integration.

Many of the students using the kit contributed components the machine building infrastructure as well as their own application. For example, to create the tendon-based whiteboard drawing robot shown at the bottom right of Figure 12, students Christoph Niess and Daniel Brun need to augment the PyGestalt library with the kinematics of their system. Their system is non-linear, which means that their transforms could not be described in matrix form. Their contribution to the control library is technically non-trivial[2].

Outside of the context of the distributed course, other machine builders have also been using the Cardboard Machine Kit to produce machines. The Cardboard Machine Kit has also been adopted into the curricula in design schools and maker spaces [18]. We will not include an analysis of these other machines in this paper, but we estimate that there are many more machines out there.

CONTRIBUTIONS

A key contribution of the Cardboard Machine Kit is demonstrating the benefits of a modular machine building approach.

[2] Niess and Brun documented their work at http://archive.fabacademy.org/archives/2016/fablabhrw/, accessed September 2016.

Type of machine	Nr observed
Plotters/drawing/painting machines	25
Hot wire cutting machines	8
3D Scanners or Animation Machines	4
Laser cutters	4
Mills or lathes	12
3D printers	2
Biology lab equipment	7
Robotic arms and 5+ DoF machines	5
Music making machines	6
Arcade game machines	5
Food preparation	18
Agricultural or solar machines	5
Sand gardening machines	2
Interactive displays	12
Other machines	10
Total	125

Table 2. Machine types that were made as part of the Fab Academy machine building projects.

By implementing motion module primitives, networked control nodes, and software to control both, we reduced the complexity of producing a digital fabrication machine.

A key insight gleaned from this modular approach is that the infrastructure of a machine and the application of a machine can be separated. Networked controls or motion primitives provide machine building infrastructure, whereas particular laser cutters or milling machines are specific applications.

By running both formal workshops and observing the Cardboard Machine Kit in the wild, we were able to confirm that this modular approach reduces the complexity of building a machine for novice and more experienced machine builders.

DISCUSSION

While we were able to confirm our hypothesis that a modular approach to machine building would enable a more diverse group of users to build machines, we were blown away but how many working machines were produced of so many different types. This is in strong contrast to our observations of users building 3D printers from open source designs.

We were very surprised at how novice machine builders were comfortable building and controlling machines which would conventionally be considered complex, such as machines with more than five degrees of freedom, or machines with vision-based closed loop control. Users could add complexity only in one part of the system without affecting other parts of the system, enabling them to develop strong contributions to the system within a single area of expertise (such as adding a new kinematic model, or creating a complex mechanical design).

Many of the machines that were produced were somewhat frivolous, such as the zen sand gardening machines. But if it is easy enough to build a machine for a frivolous purpose, it is easy to produce machines for other purposes as well. Perhaps the sand gardeners did not think their application was frivolous at all.

Modularity and end-to-end design principles were implemented in two main ways. One was modularity between hardware, software, and control electronics. Using one type of material for the frame did not preclude using particular type of control electronics, and using a particular user interface did not preclude using a particular kinematic implementation. Another was modularity within the mechanical, electronic, and software implementations—at any point a user can add a motor, a stage, or a control node. This made it easy for the machine builders add functionality on the fly.

We believe that successful modular machine building infrastructure (of which we consider the Cardboard Machine Kit a small sample) has important implications for the field of digital fabrication. Novel machines and workflows that test new applications can be constructed using these modular infrastructures. But even more strongly, this modular approach is fundamentally different from past conventions of machine building.

FUTURE WORK

The Cardboard Machine Kit introduces modularity and reconfigurability into maker-oriented digital fabrication machines. A key insight is the separation of machine building infrastructure (here the parts available in the Cardboard Machine Kit) and machine building applications (here represented by how many machine builders used the kit with custom end effectors). However, while the Cardboard Machine Kit demonstrates the efficacy of a modular machine building approach, is only one small step towards producing modular machine building infrastructure for all automation applications. To be able to accommodate machines with more extreme demands, this kind of infrastructure needs to have modules that can work under high loads and with high bandwidth. Future work will need focus on building these kinds of infrastructures for large-format, industrial, or other yet unanticipated applications.

CONCLUSION

We presented the Cardboard Machine Kit, an open and extensible toolkit for machine building. The Cardboard Machine Kit introduces inexpensive and easily modifiable modules for prototyping digital fabrication machines. Unlike conventional machine building methods, this approach encourages rapid prototyping of machines and rapid machine design iterations. The kit has successfully been deployed by novice users to build machines, both in workshop settings and in the wild.

ACKNOWLEDGEMENTS

We would like to thank all the people who enthusiastically participated in our machine building excitement. We thank How to Make (Almost) Anything students and workshop participants. We thank all the Fablabs who participated, especially the Reykjavík Fablab and Bas Withagen, the AS220 Fablab and Shawn Wallace, and the Fablab Barcelona and Ferdi Meier. We would also like to thank Fab Academy staff Sherry Lassiter, Tomas Diez, Luciana Asinari, and Jean-Luc Pierite.

Many thanks to Rob Miller, Jennifer Jacobs, Leilani Battle, Hunter Lang, and Nick Hynes for their help with framing and presenting our work in this paper.

REFERENCES

1. Morgan G. Ames, Jeffrey Bardzell, Shaowen Bardzell, Silvia Lindtner, David A. Mellis, and Daniela K. Rosner. 2014. Making Cultures: Empowerment, Participation, and Democracy - or Not?. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 1087–1092. DOI: http://dx.doi.org/10.1145/2559206.2579405

2. Ayah Bdeir. 2009. Electronics As Material: LittleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction (TEI '09)*. ACM, New York, NY, USA, 397–400. DOI: http://dx.doi.org/10.1145/1517664.1517743

3. Leah Buechley, Daniela K. Rosner, Eric Paulos, and Amanda Williams. 2009. DIY for CHI: Methods, Communities, and Values of Reuse and Customization. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 4823–4826. DOI: http://dx.doi.org/10.1145/1520340.1520750

4. David Clark. 1988. The design philosophy of the DARPA Internet protocols. *ACM SIGCOMM Computer Communication Review* 18, 4 (1988), 106–114.

5. Wei Gao, Yunbo Zhang, Diogo C Nazzetta, Karthik Ramani, and Raymond J Cipra. 2015. RevoMaker: Enabling Multi-directional and Functionally-embedded 3D printing using a Rotational Cuboidal Platform. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 437–446.

6. F. Gramazio and M. Kohler. 2014. *Made by Robots: Challenging Architecture at a Larger Scale*. Wiley. https://books.google.com/books?id=YUyPAwAAQBAJ

7. Scott E Hudson. 2014. Printing teddy bears: a technique for 3D printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 459–468.

8. Jennifer Jacobs and Amit Zoran. 2015. Hybrid Practice in the Kalahari: Design Collaboration Through Digital Tools and Hunter-Gatherer Craft. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 619–628. DOI: http://dx.doi.org/10.1145/2702123.2702362

9. J. A. Lewis. 2006. Direct Ink Writing of 3D Functional Materials. *Advanced Functional Materials* 16, 17 (2006), 2193–2204. DOI: http://dx.doi.org/10.1002/adfm.200600434

10. Evan Malone and Hod Lipson. 2007. Fab@Home: the personal desktop fabricator kit. *Rapid Prototyping Journal* 13, 4 (2007), 245–255.

11. David Mellis, Sean Follmer, Björn Hartmann, Leah Buechley, and Mark D. Gross. 2013a. FAB at CHI: Digital Fabrication Tools, Design, and Community. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 3307–3310. DOI: http://dx.doi.org/10.1145/2468356.2479673

12. David A. Mellis, Sam Jacoby, Leah Buechley, Hannah Perner-Wilson, and Jie Qi. 2013b. Microcontrollers As Material: Crafting Circuits with Paper, Conductive Ink, Electronic Components, and an "Untoolkit". In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. ACM, New York, NY, USA, 83–90. DOI: http://dx.doi.org/10.1145/2460625.2460638

13. Ilan Moyer. 2013. *A Gestalt Framework for Virtual Machine Control of Automated Tools*. Master's thesis. MIT, Cambridge.

14. Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2642918.2647359

15. Nadya Peek and James Coleman. 2015. Design Machines. In *SIGGRAPH 2015: Studio (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 2, 1 pages. DOI: http://dx.doi.org/10.1145/2785585.2792578

16. Alec Rivers, Ilan E. Moyer, and Frédo Durand. 2012. Position-correcting Tools for 2D Digital Fabrication. *ACM Trans. Graph.* 31, 4, Article 88 (July 2012), 7 pages. DOI:http://dx.doi.org/10.1145/2185520.2185584

17. Lawrence G. Roberts. 1978. The evolution of packet switching. *Proc. IEEE* 66, 11 (1978), 1307–1313.

18. Daniel Saakes. 2016. KAIST Cardboard Stage Metric. https://github.com/daan/cardboard-stage-metric. (2016).

19. Jerome H. Saltzer, David P. Reed, and David D. Clark. 1984. End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)* 2, 4 (1984), 277–288.

20. Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1277–1280. DOI: http://dx.doi.org/10.1145/2702123.2702207

21. Skylar Tibbits. 2014. 4D Printing: Multi-Material Shape Change. *Architectural Design* 84, 1 (2014), 116–121. DOI:http://dx.doi.org/10.1002/ad.1710

22. Guanyun Wang, Lining Yao, Wen Wang, Jifei Ou, Chin-Yi Cheng, and Hiroshi Ishii. 2016. xPrint: A Modularized Liquid Printer for Smart Materials Deposition. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*.

ACM, New York, NY, USA, 5743–5752. DOI: http://dx.doi.org/10.1145/2858036.2858281

23. Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. 2015. ReForm: integrating physical and digital design through bidirectional fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 93–102.

24. Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 589–598. DOI: http://dx.doi.org/10.1145/2380116.2380190

25. Willie Wu, Adam DeConinck, and Jennifer A. Lewis. 2011. Omnidirectional Printing of 3D Microvascular Networks. *Advanced Materials* 23, 24 (2011), H178–H183. DOI: http://dx.doi.org/10.1002/adma.201004625

26. Amit Zoran and Joseph A. Paradiso. 2013. FreeD: A Freehand Digital Sculpting Tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2613–2616. DOI: http://dx.doi.org/10.1145/2470654.2481361