

Tandem: Reproducible Digital Fabrication Workflows as Multimodal Programs

Jasper Tran O’Leary
jaspero@cs.washington.edu
University of Washington
Seattle, Washington, USA

Thrisha Ramesh
thrisha@cs.washington.edu
University of Washington
Seattle, Washington, USA

Octi Zhang
octipus@cs.washington.edu
University of Washington
Seattle, Washington, USA

Nadya Peek
peek@cs.washington.edu
University of Washington
Seattle, Washington, USA

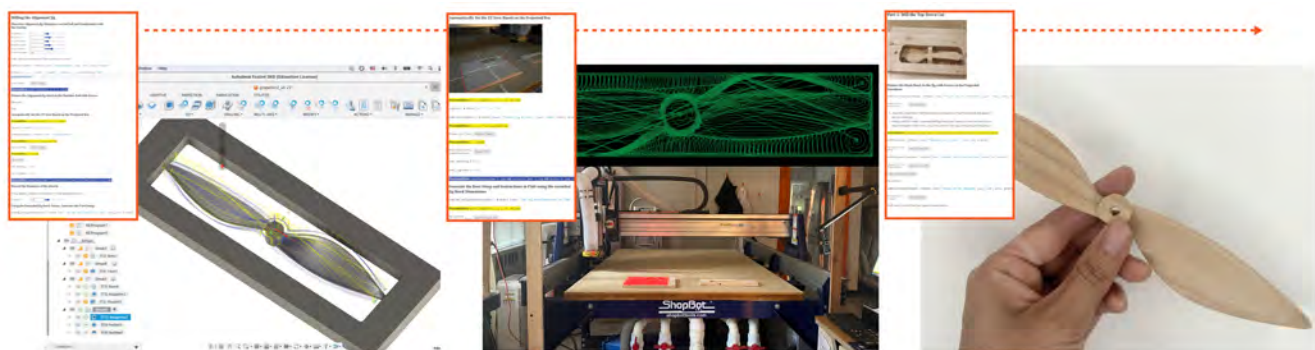


Figure 1: Tandem brings the disparate digital and physical elements needed for computer-controlled fabrication workflows together into a single program: a richly annotated computational notebook (orange).

ABSTRACT

Experimental digital fabrication workflows are increasingly common in human-computer interaction research, but are difficult to reproduce. We present Tandem, a software library that lets a fabricator implement an end-to-end fabrication workflow as a computational notebook program that others can run to physically reproduce the workflow. Tandem notebook programs read and write to CAD and CAM software, project augmented reality interfaces onto machines for manual interventions, and directly control fabrication machines. Fabricators can also denote potential mismatches between the physical and the digital as explicit assertions in code. Using two-sided CNC milling as an example, we demonstrate how to implement a complex workflow as a single program that can be re-run by others while supporting quality control and improving reproducibility.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → **Computer-aided manufacturing**; • **Software and its engineering** → **Domain specific languages**.

KEYWORDS

Digital fabrication, computational notebooks, programming languages, open-source software

ACM Reference Format:

Jasper Tran O’Leary, Thrisha Ramesh, Octi Zhang, and Nadya Peek. 2024. Tandem: Reproducible Digital Fabrication Workflows as Multimodal Programs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI ’24)*, May 11–16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3613904.3642751>

1 INTRODUCTION

From 3D printing self-supporting edible structures [82] to laser cutting fine-tuned haptic devices [85], *experimental digital fabrication* workflows feature unconventional ways of working with code, machines, and materials. These workflows string together multiple software tools, demand precise tuning of machine and material setups, and require manual intervention. In human-computer interaction (HCI) and beyond, researchers have been pioneering experimental workflows as a burgeoning topic of work.



This work is licensed under a Creative Commons Attribution International 4.0 License.

CHI ’24, May 11–16, 2024, Honolulu, HI, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0330-0/24/05
<https://doi.org/10.1145/3613904.3642751>

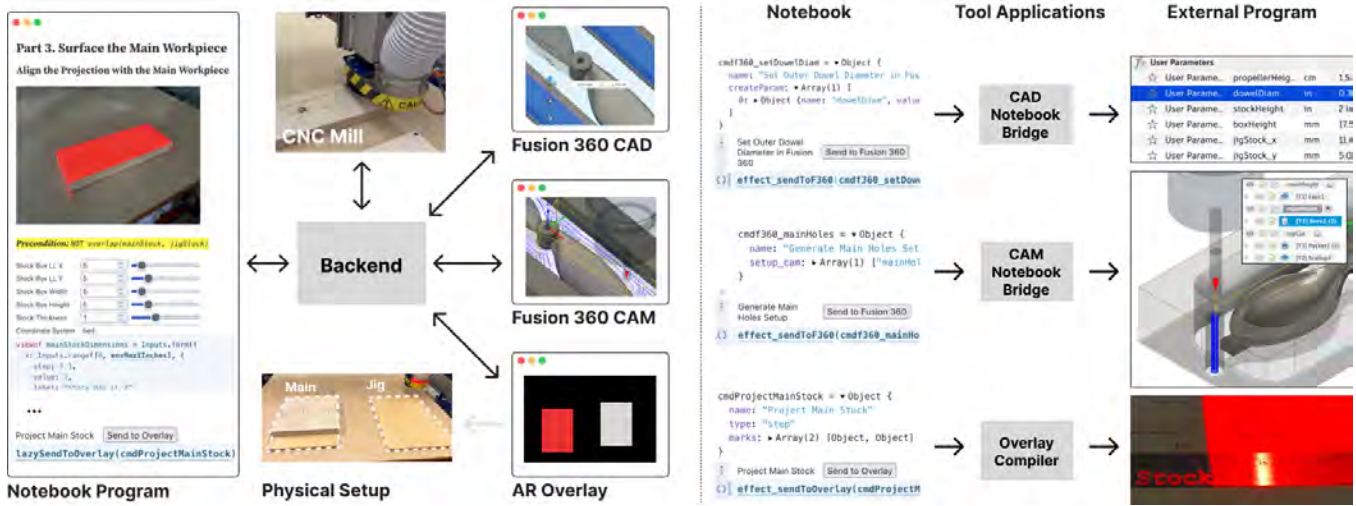


Figure 2: Tandem System Architecture. (Left) the notebook interacts with Fusion 360 CAD and CAM, with the AR Overlay, and with the CNC mill via HTTP requests. the notebook makes these requests to a dedicated backend server that forwards the requests to the respective *tool application*, e.g., an add-in we wrote for Fusion 360. The AR overlay (bottom right) prompts the fabricator to make adjustments to the physical setup. (Right) Tandem’s API communicates over HTTP with tool applications in the notebook. Top: sending a value calculated in the notebook to be set as a user parameter in Fusion 360 CAD; Middle: automatically generating a toolpath in Fusion 360 CAM to mill alignment holes; Bottom: rendering boxes on the mill’s bed at specified positions using the AR overlay.

Yet, despite the rise of experimental digital fabrication in HCI, we are missing a robust way to share working implementations of research with others. In contrast, in fields like the basic sciences, reproducibility of past research is fundamental [2, 50, 55]. Researchers and practitioners should be able to reproduce, validate, and build upon prior work. In the case of experimental fabrication, those who would benefit from reproducibility are not only novices to digital fabrication, but also experts unfamiliar with a newly invented fabrication workflow.

To illustrate this scenario, consider two personas separated by geographical distance: Florence, a community wood shop manager who is experienced with CNC milling; and Siena, an HCI researcher who is working on new interaction techniques for bringing craft-based techniques to CNC mill control. Florence has developed a workflow for doing *two-sided milling*, where a fabricator carefully mills on two opposite sides of a workpiece using a conventional 3-axis mill. Siena is eager to learn two-sided milling not only to increase the types of objects she can manufacture on her CNC mill, but also to apply techniques used in Florence’s workflow to her own research.

However, there is currently no way for Florence to create reproducible versions of her novel digital fabrication workflows beyond writing a tutorial—a helpful but incomplete representation. When Siena follows text or video tutorials (e.g., [3, 25, 34, 44, 74]), she must manually re-implement every step of the workflow using her own software tools and machine hardware. Imprecise re-implementation can result in errors that are time-consuming and difficult to debug because Siena needs to reason about interdependent physical and digital states. Figure 3 shows example errors that arise in two-sided

milling. In contrast, with digital-only programming, Siena can easily fork repositories, run the code to get an immediate result, read parts of code in depth, and experiment with her own changes.

In particular, three challenges stand out for Siena in replicating Florence’s workflow.

- (1) Florence uses several disparate software tools interdependently, including conventional CAD/CAM software. There is no single environment to present a linear sequence of steps for Siena to execute in code.
- (2) Siena must adjust physical machine and material settings to match assumptions made in digital code, or vice-versa; it is difficult for Siena to know when to manually intervene and how to do so.
- (3) Each step of Florence’s workflow has implicit preconditions about what must be satisfied before running the mill. It is difficult for Florence to explicitly formulate these “gotchas” so that Siena does not repeat the same mistakes Florence encountered.

To provide an infrastructure for implementing reproducible versions of novel fabrication workflows, we present Tandem, a library and backend that lets fabricators like Florence implement their entire workflow as a computational notebook program (Figure 1). Florence uses Tandem’s API to author her workflow as function calls to CAD and CAM software, to an *augmented reality* (AR) overlay for manual interventions, and to the CNC mill itself (Figure 2). Tandem also lets Florence include *assertions* in her code that check preconditions that she provides before fabricators like Siena execute error-prone milling steps. Fabricators like Siena then reproduce the workflow simply by executing the code, following instructions that



Figure 3: Examples of Errors that Arise in Two-Sided Milling. (1) A milled alignment hole is too narrow for the dowel to fit, requiring a second set of holes to be milled. (2) The CAD design of a spoon features a handle that is too narrow and gets torn off by the mill. (3) Flipping the workpiece over an axis that deviates too much from an assumed axis of rotation results in an invalid post-milling geometry. (4) An error taken from an online forum [64] where a fabricator reported misaligned features while milling a guitar. Another forum user responded that the reason was that the alignment holes were asymmetric across the axis of rotation assumed by the original poster.

the notebook produces, and manually intervening as directed by the AR overlay; Siena may also interact with the code itself if she wishes to adjust parts of the workflow.

We evaluate the feasibility of Tandem by implementing an entire end-to-end two-sided milling workflow as a computational notebook program and producing four artifacts as a result (Section 7). We chose two-sided milling as an example novel fabrication process because it is more difficult than many; it involves tool changes, synchronizing machine and material setups in physical and digital states, measurement and manual inspection, and experimenting with milling feed and speed. We elected to explore this single workflow in depth rather than explore multiple digital fabrication workflows at a shallower level. While Tandem’s features are most clearly showcased in two-sided milling, they can be applied to many other novel fabrication processes as well, including 3D printing, laser cutting, and laboratory automation (see subsection 8.3).

The structure of the paper is as follows: after describing related work, we first walk through how a fabricator who reproduces a workflow (Siena) uses a Tandem notebook; then, we walk through how an authoring fabricator (Florence) implements the workflow’s CAD, CAM and machine control functionality using Tandem’s API (Section 4). We then detail how Florence programs physical steps using the AR overlay (Section 5), and how Florence guards against a large class of easily preventable errors using assertions (Section 6). Finally, we describe the demonstration artifacts fabricated with the notebook alongside artifact-specific assertions (Section 7) before concluding with lines of future work that are now possible with a reproducible fabrication workflow (Section 8).

2 RELATED WORK

2.1 Replicating Experimental Digital Fabrication in HCI

We built Tandem to support the burgeoning body of work in human-computer interaction that studies new ways of using digital fabrication machines. For example, experimental additive manufacturing workflows include: printing with Play-Doh [4], printing garments via selective underextrusion [15], printing a wireframe of an object

[46], 3D printing of editable materials [82] and translating audio data to printed ceramic ware [10]. In the last case, the authors noted the difficulty of explicitly representing the “how-to” of machine-material negotiation, writing that the necessary “close relationship with the machine was possible because [Author 2] already had an excellent understanding and anticipating of what the printer could do, including the innovative use of the dwell command for the production of texture.” These comments showcase how experimental workflows make assumptions about the physical setup that are difficult for future fabricators to implement. Not every paper includes the necessary details to navigate these assumptions, and, even in those that do, it remains difficult for other fabricators to translate experimental details into working code.

Experimental workflows, and their potential for physical-digital mismatches, also encompass subtractive manufacturing workflows other than two-sided CNC milling. For example, Zheng et al. [85] presented a technique for producing small haptic devices using only a laser cutter and POM material. While the authors developed a computational design sandbox for developing swatches of mechanisms, swatch production is specific to Zheng et al.’s particular workflow. There is no simple way to generalize this approach to different materials, machines, or toolpath choices. The same issue holds true for fabrication techniques that involve stencil-based production, for example, with biodegradable conductive materials [31] and conductive ceramic ware [84]. Subtractive workflows for custom wood joinery [35] can fail if the corresponding digital tool does not model any unpredictable behavior with the milling machine.

Tandem aims to support such existing work in experimental fabrication and help generalize the material- and machine-specific constraints that authors have already explored in their respective workflows. By documenting both computational functionality and necessary human interventions, Tandem helps fabricators make their work more reproducible by others [2, 55].

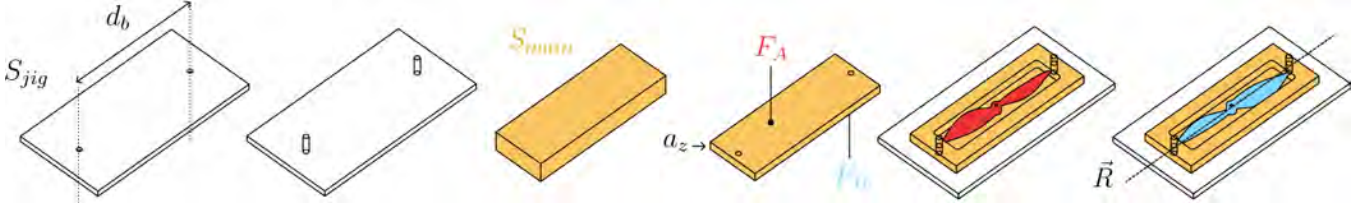


Figure 4: Schematic of Two-Sided Milling a Propeller in Tandem. The fabricator, Siena, mills alignment holes in a sheet of wood and inserts two dowels to form the *alignment jig* S_{jig} (white). Moving to the *main workpiece* S_{main} from which the propeller will be milled (yellow), she mills the S_{main} down to exactly a_z in height and mills two additional alignment holes. We denote the top and bottom faces of S_{main} as F_A (red) and F_B (blue), respectively. Next, she fixtures S_{main} on top of S_{jig} and mills F_A in two passes: a roughing pass and a finishing pass (not shown). Finally, flipping S_{main} over the predetermined axis of rotation $\vec{R}$, Siena mills all areas reachable from the bottom of the propeller into F_B , again in two passes.

2.2 Programming Tools for Fabrication

While many existing tools support programming for digital fabrication, few address concerns in experimental workflows that span digital and physical.

Most programming tools for fabrication focus solely on the CAD stage of a workflow, spanning domains such as carpentry [79], design enclosures for electronic devices [27], patterned designs for several types of fabrication processes [24], and countless others. Research in programming languages has targeted CAD, for example, by decompiling meshes to parametric CAD [49], improving language constructs for parametric histories [6], and generating designs via reinforcement learning over CAD datasets [77]. However, these works operate wholly within the digital and not the physical sphere. In contrast, we contribute programming language constructs that tie CAD-level designs to physical concerns. A smaller area of work has investigated structured direct interaction with fabrication machines such as lathes [69], laser cutters [47], large format CNC drawing [39], and sheet material manipulation [70]. In particular, Piovarči et al. [56], showed how to develop reinforcement learning algorithms to improve toolpath design using data observed from a camera. Overall, these examples are workflow-specific and

do not generalize to other cases. Tandem brings material-aware logic of these examples to a single programming environment.

Few prior works feature full-fledged programming environments that handle material and machine behavior at all. One example by Fosdall et al. [17, 18] integrates machine and CAM actions within the Grasshopper visual programming environment [45]. Other works use Excel [19] or creative Javascript coding environments [36, 65]. Tran O’Leary et al. showed how computational notebooks, normally used for digital-only applications, could be extended to provide a versatile programming environment for fabrication [71]; we build on initial concepts explored in this work to form Tandem’s core programming environment. Compared to these works, Tandem builds on existing programming environments by adding integration with conventional CAD and CAM tools, adding integration with augmented reality guidance, and adding assertions to align code with empirical machine and material conditions.

2.3 Physical-Digital Guarantees Outside Fabrication

Most previous tools for experimental fabrication, be they one-off workflows or more extensive CAD, CAM, or programming tools, lack explicit ways to negotiate assumptions made in code with physical realities. However, outside of digital fabrication, researchers have indeed been representing physical factors as “first class” concepts that can be used in code [22]. For example, Vrana et al. [75] proposed a system, Aquarium, that lets scientists program a wet laboratory workflow both as a high-level flow chart and as a user-facing step-by-step interface that technicians follow. We were particularly inspired by Aquarium’s domain-specific language for building user-facing workflows that support warnings, preconditions, and links to physical reagent locations and equipment in the lab. In the same domain, Willsey et al. [78] built a language to implement lab protocols on a microfluidics machine. The system provides error correction through the use of computer vision; if the physical configuration of droplets on the machine deviate from an expected state, the system creates a new empirical program state and rolls back future commands.

Apart from laboratory automation, formal methods can also guide crafters without expertise in creating foundation-pieced quilts [37]. Similarly, they have been used extensively to verify VLSI chip

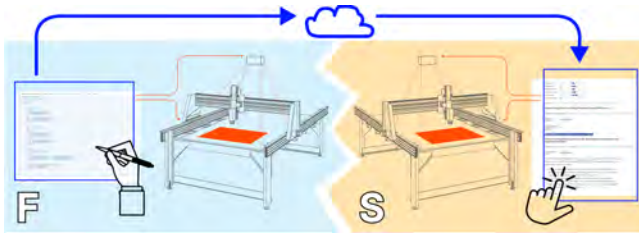


Figure 5: Programming and Handing Off a Notebook. Left) Florence refines her two-sided milling process on her own. She then implements the steps of her process in notebook cells by writing source code with Tandem’s library calls. Right) Siena navigates to the notebook in the browser, starts the Tandem backend on her laptop, and connects it to the projector and CNC mill. She works through the notebook to replicate Florence’s workflow.



Figure 6: Moments from Siena’s Steps in the Notebook. (1) Given the already modeled spoon, Tandem generates the outer and tabs that surround the spoon. (2) A completed alignment jig. (3) The main workpiece S_{main} after being milled to the height of the spoon a_z . (4) The top face F_A of the main workpiece S_{main} after milling the top-down cut. (5) The bottom face F_B of the main workpiece S_{main} after milling the bottom-up cut.

designs [62], robot task specifications [32, 33], and interactive programming by demonstration with robot arms [59]. In general, we observe existing techniques to write programs that synchronize steps in code with physical requirements in laboratories, with quilting, or with robots. Such techniques have yet to be applied to digital fabrication tasks. Tandem is a first step towards reflecting machine and material states, alongside human input, in code itself.

2.4 Augmented Reality for Fabrication

One goal of Tandem is to help fabricators adjust physical circumstances, e.g., adjust the physical setup on a milling machine or change the position of the machine’s tool. Tandem does so by providing a grammar of AR-based interactions that guides the fabricator to carry out these physical steps (see Section 5). This functionality complements an existing body of work in fabrication and human-robot interaction (HRI) that leverages AR, for example: machine maintenance with AR [13, 67], CNC milling with a camera feed that supports direct manipulation [48], AR-supported collaborative making with a robot arm [68], and projecting CNC machine parameters onto machine safety glass [52].

Outside of CNC milling, many HRI papers leverage AR to enhance interactions between humans and robots; Suzuki et al. [66] created a taxonomy of several design dimensions for this emerging space. One focus area is helping people work with understanding [29] and debugging [7] printed circuit boards by projecting data from schematics onto the physical boards. Follmer et al. [14] and Weichel et al. [76] both use AR to prototype interactions for returning physical 3D scan data back to a programming interface. Mahapatra et al. [43] further study pain points that fabricators face when working with data from existing physical constraints. AR methods have also been leveraged for mixed-reality 3D modeling [40], 3D printing [12, 54, 80], carving [20, 21, 86], laser cutting [47, 57], building-scale construction [83], and for providing tutorials for fabrication machines [23, 60, 81].

These works prototype fixed interactions between a machine and an AR interface. In contrast, Tandem uses programmable AR interfaces within a coding environment—fabricators use AR interactions to specify what other fabricators must do to accurately replicate a given workflow.

3 WALKTHROUGH: REPLICATING AN EXISTING WORKFLOW

We now provide a high-level walkthrough of how a fabricator like Siena would follow Florence’s notebook to replicate Florence’s workflow; her goal is to CNC mill a spoon (Section 7). Section 4 provides a code-level explanation of how the workflow is implemented in part of the notebook.

To start the walkthrough, Siena would need to have installed Tandem’s backend on her computer; then, she opens an Observable computational notebook [51] in her web browser; the notebook contains all the source code, prose, and input elements that she uses to interact with CAD/CAM tools and an AR overlay for interacting with the physical setup of materials on the mill’s bed and with the mill itself. Note that our implementation provides a messaging service for the backend to communicate with the Shopbot PRSAlpha Mill [63] that we use; to use another machine instead, the fabricator would need a service built for that machine or custom built if one does not exist.

Once the notebook is open, Siena proceeds through the 6 notebook sections, which we represent in respective subsections below. We present the high-level progression of these steps in Figure 4 and moments corresponding to each step in Figure 6.

3.1 Prepare the CAD File

Siena opens the Fusion 360 CAD file of the spoon she wants to mill in Fusion 360. While there are several approaches to two-sided milling, Tandem’s current approach uses two *alignment dowels* that hold the *main workpiece*, i.e., the piece of material being milled into a spoon, in the same place after Siena flips it over to mill the second side. This means that Siena must add an extra part to her CAD file, which we call the *outer*. The outer, pictured in Figure 6 Part 1, contains two precisely positioned holes for the alignment dowels and connects to the spoon itself using tabs. Using the notebook, Siena enters the diameter of the dowels she measured and the notebook automatically generates the outer in Fusion 360 CAD.

3.2 Mill the Alignment Jig

Next, Siena mills the *alignment jig*, which holds the main workpiece of the spoon in place while it is being milled (Figure 6 Part 2). Using the AR overlay, the notebook guides Siena through cutting down and positioning the *jig workpiece* from which she will mill the jig. The overlay projects a rectangle onto the the CNC mill’s bed where

the stock material goes. Once the physical stock is synchronized with the projection, the notebook changes the overlay to a new projection that shows Siena where she should drill screws to fixture the stock to the mill’s bed. Using the dimensions gathered about the jig workpiece, the notebook then generates a toolpath in Fusion 360 CAM for the machine to mill the alignment holes in the jig workpiece. It then guides Siena through manually setting the zero (the XYZ origin) on the CNC mill to match the zero assumed by the toolpath in CAM. Finally, the notebook generates low level machine instructions for the toolpath and lets Siena dispatch the instructions to the mill directly from the notebook. After the holes have been milled in the jig workpiece, Siena presses the dowels into place to complete the jig.

3.3 Mill the Main Workpiece to the Correct Height

Next, Siena moves to the main workpiece and mills it down to the exact height of the spoon that will be milled. She mostly repeats the same steps she took with milling the alignment jig except for using the main workpiece. However, because the mill will remove all material above a certain height, she must be careful to screw the main workpiece to the bed so that the screws lie below the lowest cut that the mill will make. To do this, Tandem uses *counterbores* that position the top of a screw lower than the top of the material that it is screwed into, as shown in Figure 6 Part 3 post-milling. The notebook calculates the depths and locations of the counterbores that Siena needs to make and projects a visual guide using the AR overlay (Figure 12 Right).

3.4 Mill Alignment Holes in the Main Workpiece

Once the main workpiece is milled to the exact height of the spoon, Siena works through steps in the notebook to mill alignment holes in the main workpiece. This is the same as milling holes in the alignment jig, except applied to the main workpiece.

3.5 Mill the Top-Down Cut (Face A)

After the four previous preparation steps, Siena is finally ready to mill the spoon itself. The notebook guides Siena through otherwise error-prone steps. First, the notebook generates toolpaths in CAM that mill the main workpiece from the “top-down” before Siena flips the workpiece on the alignment jig and mills “bottom-up” from the perspective of the workpiece. Next, the notebook asks Siena to establish a designated axis of rotation $\vec{R}$ around which she will flip the part. the notebook adjusts the origins in CAM to match this axis. Finally, the notebook prompts the AR overlay to preview the cut before milling so that Siena can visually confirm that the cut looks correct before dispatching it from the notebook; the workpiece then looks like that shown in Figure 6 Part 4.

3.6 Mill the Bottom-Up Cut (Face B)

Now, Siena has milled the spoon completely from the top downwards, but she must flip it over to mill the parts that were unreachable with the workpiece’s first orientation. To this end, the AR

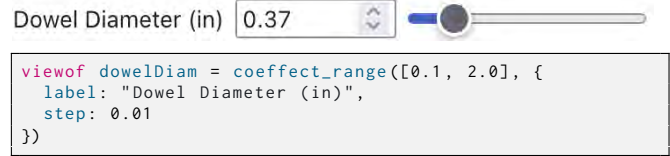


Figure 7: Prompting for User Input with a Coeffect Function. Some properties are omitted in the listing for brevity.

overlay prompts Siena to flip the workpiece in this direction according to $\vec{R}$, which was set previously. Siena then mills the bottom-up cut (Figure 6 Part 5), removes the main workpiece from the mill, and separates the spoon from the tabs, concluding the workflow.

4 IMPLEMENTING A WORKFLOW USING TANDEM

To better describe how Florence uses Tandem’s API to implement her workflow as a notebook for Siena to use, we walk through the code for the first two sections of the notebook¹: preparing the CAD model and milling the alignment jig.

4.1 Prepare the Model in CAD

In our implementation, the notebook communicates, for both CAD and CAM functionality, with Autodesk Fusion 360. We assume that Siena has already modeled the object she wishes to fabricate in Fusion 360. Then, she uses the notebook to talk directly to Fusion 360 and make necessary adjustments for two-sided milling. For example, the notebook prompts her to measure the diameter of two dowel to be used to hold the workpiece in place during milling. To do this, the notebooks uses a *coeffect function*, which generates a user input element; when Siena types in the diameter or adjusts the slider, that value is captured in the variable `dowelDiam`, as shown in the notebook snippet in Figure 7. Since CAD and CAM state is stored in Fusion 360, Siena can still make adjustments in Fusion 360 and re-run any necessary effect functions from the notebook.

For all notebook snippets, the *notebook cell*, which contains the source code, appears below the result of evaluating the code. In this case, the `coeffect_range` function generates the slider element above and assigns the current value of the slider to `dowelDiam`.²

Then, in Figure 8, Siena sends the diameter of the dowel to Fusion 360 to be set as a *user parameter*, which is a value that Fusion 360 can reference to set the actual diameter of the hole in CAD, mutating the model accordingly. To do this, she uses an *effect function*, i.e., a function that generates a button that, when pressed, sends the value to Fusion 360. Effect functions send messages to Fusion 360 CAD and CAM, to the AR overlay, or to the CNC mill—we refer to these recipients as *tool applications*. They take *commands* as arguments, which contain parameters that dictate what the receiving tool application should do; for example, Figure 12 shows commands for changing the AR overlay. We generate a button that must be

¹The source code for the entire notebook and the Tandem backend can be found at <http://depts.washington.edu/machines/projects/tandem/>.

²For those familiar with Observable Notebook’s input functions, `coeffect_range` is an alias for `Inputs.range`.


```
cmdf360_setDowelDiam = ▼Object {
  name: "Set Outer Dowel Diameter in Fusion 360"
  createParam: ▼Array(1) [
    0: ►Object {name: "dowelDiam", value: 0.37, unit: "in"}
  ]
}
```

```
cmdf360_setDowelDiam = ({
  name: "Set Outer Dowel Diameter in Fusion 360",
  createParam: [{
    name: "dowelDiam",
    value: dowelDiam,
    unit: "in"
  }]
})
```

Set Outer Dowel
Diameter in Fusion
360

Send to Fusion 360

```
effect_f360(cmdf360_setDowelDiam)
```

Figure 8: Mutating an External CAD Model with an Effect Function.

Workpiece X	13.6	<input type="range"/>
Workpiece Y	5	<input type="range"/>
Workpiece Width	10	<input type="range"/>
Workpiece Height	5	<input type="range"/>
W. Thickness	0.7	<input type="range"/>
Coordinate System	bed	

```
viewof jigWorkpieceDimensions = coeffect_form({
  x: coeffect_range([0, envMaxXInches], { ... }),
  y: coeffect_range([0, envMaxYInches], { ... }),
  ...
})
```

Figure 9: Combining Multiple Coeffect Functions with coeffect_form. The function evaluates to an object containing the values set on the sliders as values representing the contingent position and dimensions of a workpiece.

pressed first, rather than simply sending the value, because Observable Notebook is a live programming environment that frequently re-evaluates cells when related cells change. Thus, eagerly sending the values would result in extraneous messages being sent.

Once Siena presses the button, Tandem knows that the model in Fusion 360 has the correct user parameter for the alignment holes. Afterwards, the notebook walks her through several more steps, including: generating the *outer*, which bounds the artifact to be milled; generating the alignment holes in the outer; and generating tabs that connect the outer to the artifact. At the end of this notebook section, the CAD model is ready to be transformed through CAM into *toolpaths* that dictate the mill’s motion.

4.2 Mill the Alignment Jig

The alignment jig consists of a flat sheet of wood (the *jig workpiece* S_{jig}) with two dowels that are press-fit into respective alignment

```
cmdProjectJigWorkpiece = ▼Object {
  name: "Project Jig Workpiece"
  type: "step"
  marks: ▼Array(1) [
    0: ►Object {type: "box", location: Object, width: 10, height: 5}
  ]
}
```

```
cmdProjectJigWorkpiece = ({
  marks: [{
    type: "box",
    location: {
      x: jigWorkpieceDimensions.x,
      y: jigWorkpieceDimensions.y
    },
    width: jigWorkpieceDimensions.width,
    height: jigWorkpieceDimensions.height
  }],
  ...
})
```

Project Jig
Workpiece

Send to Overlay

```
effect_sendToOverlay(cmdProjectJigWorkpiece)
```

Figure 10: Using Position and Dimensional Information to Generate a Command. When sent to the AR overlay, the command generates a rectangular projection on the machine’s bed corresponding to the values provided.

The jig workpiece
is aligned with the
projection. ☐

```
viewof jigWorkpieceCorrect = coeffect_toggle({
  label: "The jig workpiece is aligned with the  
projection.",
  validator: jigWorkpieceDimensions
})
```

Figure 11: Coeffect Function for Checking that the AR Overlay’s Workpiece Projection Aligns with the Physical Workpiece.

holes. To properly generate the toolpath for the alignment jig, Siena must measure the physical material’s dimensions and location.

To align the physical dimensions and position of S_{jig} with code in the notebook, Siena uses a coeffect function (sliders) to generate a contingent list of dimensions and position on the mill bed (Figure 9).

In Figure 10, she then uses the gathered data to construct a *command* for the AR overlay (more information in Section 5). Next, she passes the command to an effect function to project a rectangle of light corresponding to the contingent dimensions via the AR overlay, as shown in Figure 12 Photo 1.

Siena iterates by changing the dimensions on the sliders and/or repositioning or cutting down the jig workpiece. After this iterative process, the AR projection, representing the dimensions sent from the notebook, matches the top of the physical jig workpiece. Siena then checks a box generated by a coeffect function (Figure 11) so that the function now evaluates to true; this result can subsequently be used in an assertion (see Section 6).

Once the dimensions and position in the notebook are matched with the physical setup, the rest of the section generates a CAM setup with this information, helps Siena zero the tool according to the CAM setup, fixtures S_{jig} , generates a toolpath from the setup, and dispatches the cutting job.

5 THE AR OVERLAY

The preceding programming examples show how Florence programs the AR overlay to let Siena “read from” and “write to” the state of the physical setup. Through a declarative syntax, Florence writes code to help Siena position a workpiece, drill fixture screws at designated points, double-check machine zeroes, and more. Specifically, the AR overlay is an interface projected from a projector mounted above the CNC mill onto the mill’s bed. We implemented the AR overlay as a standalone web page whose appearance changes based on *overlay commands* (see below) sent from the notebook.

5.1 Overlay Grammar

Siena interacts with the AR overlay by sending commands in JSON syntax to Tandem’s backend. A command consists of projected *marks*, which help her complete a step in the notebook. A mark could be an outline of where she should place the stock, a crosshair indicating where the center, or zero, is, or a more complex visualization, such as the toolpath of certain steps, like surfacing the stock. For each step in the notebook, Siena can send a group of marks to the projector using the “Send to Overlay” button, which appears after calling the respective effect function. The overlay and notebook together provide visual and text directions on how to accurately complete each step needed for the workflow.

The code snippets included in Figure 12 show the syntax fabricators use to create different overlay commands to send to the projector. Florence specifies several properties in the command for Siena, e.g., the type of mark, mark location, and arguments to pass to the overlay. For example, for the `screwDepth` annotation, stock thickness and model height are both needed to calculate a safe drilling depth.

5.2 Toolpath Visualizations

One important mark for overlay commands is *toolpath*, which projects a visualization of the toolpath for a certain step in the milling workflow. For example, Figure 14 shows an overlay for milling the main workpiece down to the correct height. We implemented the “toolpath” mark by extending techniques proposed in previous work on AR toolpath visualization [72]. Figure 13 shows two overlays corresponding respectively to the roughing and finishing passes for F_A for the propeller.

5.3 Calibrating the Overlay

To calibrate the overlay for her machine-projector setup, at the beginning of the notebook, Siena projects a calibration rectangle onto the CNC mill’s bed. She sets the position and dimensions of the rectangle to match a ground truth rectangle on the mill’s bed; the rectangle should encompass the area in which she will place workpieces. Siena then drags the distorted projected rectangle’s corners to match the ground truth rectangle on the bed. From these

four point transforms, Tandem computes a homography that ensures that marks appear at the correct x and y location on the bed at depth $z = 0$. For greater depths, i.e., marks projected on a workpiece, we downscale the projection by a factor of $1 - \frac{\theta}{d_{lens}}$ where θ is the thickness of the workpiece and d_{lens} is the distance from the projector’s lens to the machine bed. With this relatively simple calibration technique, we found that test marks projected were precise within 3mm of the specified position. More sophisticated techniques could reduce this margin of error.

6 PHYSICAL DIGITAL ASSERTIONS

When implementing her workflow in Tandem, Florence wants to prevent future fabricators like Siena from encountering the same errors she did. As with any novel fabrication workflow, errors arise from many sources: digital logic, material behavior, artifact geometry (CAD), manufacturing settings (CAM), and setups on the physical machine. Further, many errors arise from mismatched assumptions between these disparate factors. For example, CAM software may assume a given coordinate space origin (the *zero*), but the zero assumed by the fabricator in their physical setup is different (see Figure 16). Errors with novel workflows can range from being time-consuming at best to catastrophic at worst.

To solve this problem, Tandem lets workflow implementers like Florence write *physical-digital assertions* in the workflow’s code. In digital-only programs, assertions check whether a *predicate* evaluates to true, and, if not, typically halts program execution; for example the Python statement `assert material_thickness > 0` will raise an exception at runtime if the variable `material_thickness` is less than zero. Assertions are available in most modern programming languages, including some end user programming environments [5]. The goal of physical-digital assertions is to extend the reach of assertions from digital-only code to now include fabrication workflows.

6.1 Example Assertions

We briefly walk through examples of how Florence notices where she can write assertions and how she does so in Tandem. As a simple example, Florence wants to let fabricators know to install a quarter-inch flat end mill before milling the alignment jig. Intuitively, Florence wants to make sure that Siena and any other fabricators using the notebook mill the alignment jig only if they have first installed the correct tool. Equivalently, we can say that the code that mills the jig executes only when a prior tool installation assertion evaluates to true.

As basic tool change assertion, Florence writes her precondition as a predicate in Javascript code as shown in Figure 15. As with other calls to `coeffect_toggle`, the notebook generates a checkbox with a label instructing Siena to check the box only if she has manually changed the tool to the specified end mill. Because `coeffect_toggle` evaluates to true if the checkbox is checked and false otherwise, `mainEndMillInstalled` is a valid predicate. In an adjacent notebook cell, Florence can also describe in prose why her precondition is necessary, e.g., to prevent leftover material that can result from drilling the jig with a ball end mill.

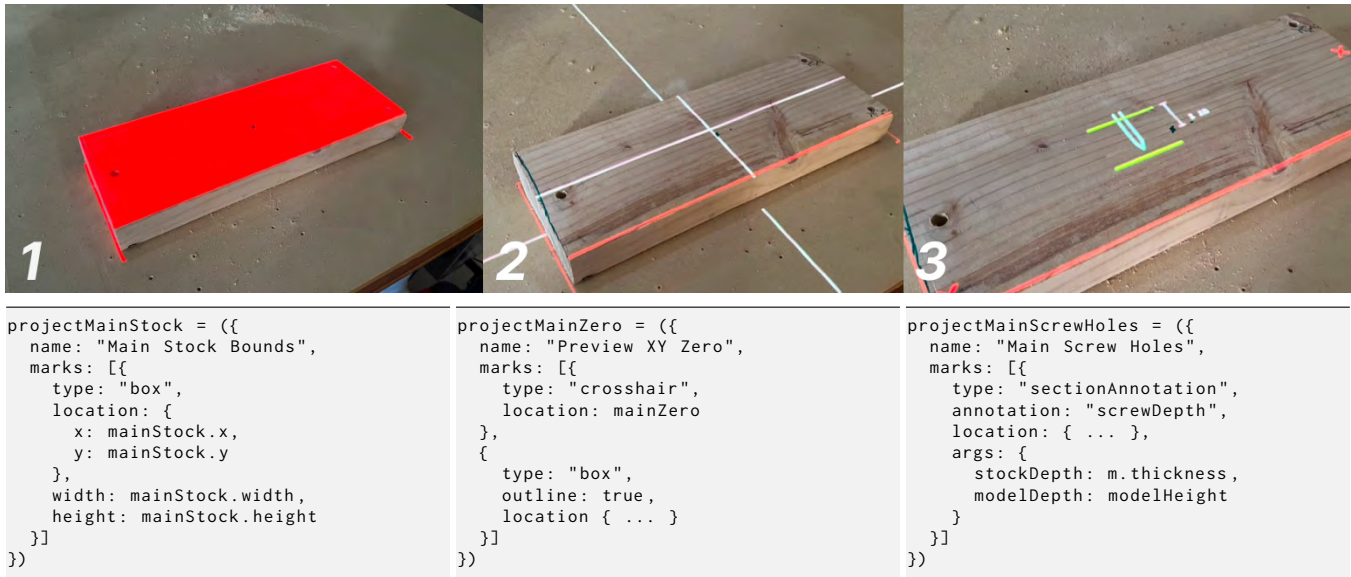


Figure 12: Example Overlay Commands on the Physical Setup (top) and the Command Syntax (bottom). (1) Projecting a red rectangle with the same dimensions as the stock helps the user place the stock correctly on the bed before milling. (2) Projecting a crosshair at the center (or zero) of the stock helps the user zero the machine at the center of the stock, as assumed in a corresponding CAM setup. (3) The screwDepth annotation generates lines that show the top/bottom surfaces of the stock and drill bit as well as a label that indicates how deeply the user must drill the screws when securing the stock to the bed before milling.

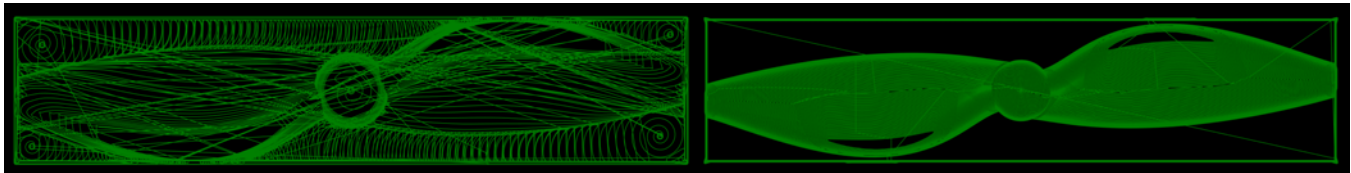


Figure 13: Toolpath Visualizations for the Roughing (left) and Finishing (right) Milling Passes for F_A of the Propeller.

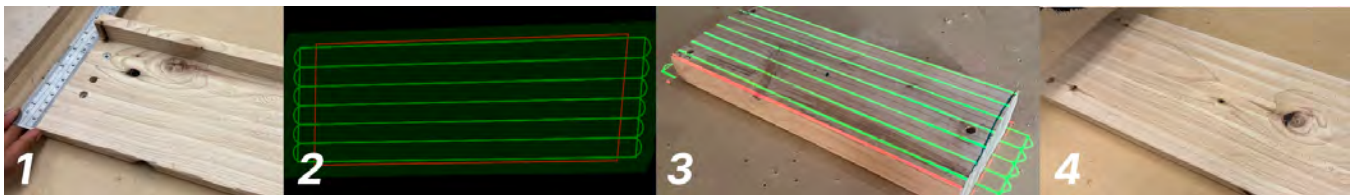


Figure 14: Preventing Leftover Stock while Milling the S_{main} to the a_z . (1) Slight deviations in the physical position of S_{main} , versus the ideal position, can cause leftover "slivers" of unmilled stock. (2) A screenshot of the web interface for the AR overlay with a visualization of the toolpath for cutting down to a_z . The red boundary is the calculated boundary of S_{main} , and the green lines represent the path of the end mill. The fabricator adjusts the "stock offset" option in CAM to increase the bounds of the green lines beyond the red boundary. Too little offset results in unmilled stock, while too much results in wasted time spent milling air. The entire rendering in the web interface is warped using an empirically gathered perspective transform to account for the positioning of the projector above the machine. (3) The overlay projected onto S_{main} . (4) S_{main} after being milled with the corrected toolpath.

Florence then passes `mainEndMillInstalled` as an argument alongside other predicates to the `requires` function which evaluates to true exactly when all the predicates evaluate to true. As shown in Figure 17, the result of `requires` is thus a “checklist” of preconditions that must be fulfilled before Siena can run the jig milling step on the physical machine. She passes the result to the effect function `effect_sendToMill`; if `precondFinishing` evaluates to false, then the button generated by `effect_sendToMill` will be disabled to prevent error-prone milling.

Altogether, combination of `requires` and an effect function comprises a physical-digital assertion. Florence then repeats the process for each milling step in the notebook, first writing predicates that are specific to each step, then combines them using `requires` whose resulting value she passes to each effect function.

As a more complicated example, consider the final predicate in Figure 17, “CAM zero matches physical zero.” This predicate means that the origin used in the CAM setup—where the point (0, 0, 0) is assumed to be in relation to the stock material—must match the true physical location at which the machine was zeroed (see Figure 16). To zero the machine, Siena must first move the tool head to the desired location and then run the zeroing command on the mill, which sets the location to (0, 0, 0). This establishes a local coordinate system relative to the stock material, while the world coordinate system remains relative to the unchanging machine bed.

To implement this predicate, Florence must check that the origin used in the CAM setup visually matches the zero set in hardware using the overlay and user confirmation. To do so, she writes an effect function to safely move the mill’s tool to the zero currently in hardware. The notebook then prompts fabricators like Siena to visually compare the current position of the tool in hardware with the zero assumed in the CAM setup; Siena checks the predicate’s checkbox to confirm a visual match.

6.2 Scope and Limitations of Assertions

The goal of assertions is to allow Florence to prevent a large set of predictable errors from occurring when Siena uses the notebook by explicitly including checks and preconditions in the notebook’s code. Just like in digital-only code, assertions cannot prevent all errors. In Tandem’s case, assertions focus on errors that are easily preventable as long as Siena is made aware of prerequisite steps such as mismatched CAM zeros, tool changes, and material positioning. While these checks may be “trivial” to those experienced with a given workflow, they are easily missed as workflows grow more complex or deviate from techniques that Siena is used to.

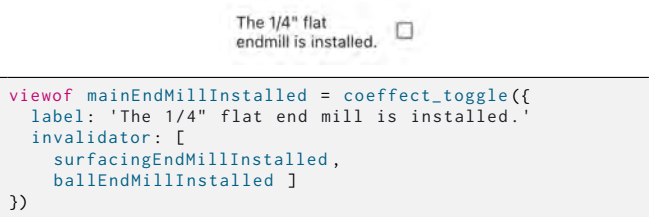


Figure 15: A basic predicate that prompts the fabricator to manually check that a given end mill is installed.

Assertions can be more or less useful depending on their implementation in code. For example, Florence could improve the simple checkbox that Siena checks to verify that the correct tool is installed. Namely, she could implement a model that uses computer vision to classify the currently installed tool³. In addition, rather than using a hard-coded end mill, Florence could query the current CAM setup’s designated tool using a `coeffect` function. Currently, predicates primarily rely on user verification such as checkboxes to “sense” physical state; this suffices for many types of assertions we encountered. Still, in future work, we intend to leverage sensor data, such as video feed, touch sensors, and audio data to automatically satisfy or invalidate assertions based on empirical data (see subsection 8.2).

Finally, assertions are not meant to be absolutely binding. Like any programmer, Siena can check off, rewrite, or remove an assertion if she finds her own working style is at odds with an assertion that Florence implemented. Even in this case, Florence’s assertion still serves as a “heads up” and Siena can add the assertion back in later if needed. Also, we note that physical-digital assertions are *not* the same as design rule checks that occur solely within CAD or CAM, e.g., checking for collisions while simulating a toolpath. Instead, these assertions concern potential errors that cross *between* physical setup and CAD/CAM, such as synchronizing the zero in CAM with the zero set on the machine.

7 DEMONSTRATIONS

To demonstrate Tandem, we (the authors) used our two-sided milling program to mill four artifacts: a propeller, a spoon, a bowl, and a printed circuit board (PCB). As we fabricated each artifact, we documented the challenges we faced, e.g., the spoon snapping during a roughing operation. To prevent other fabricators from encountering these same errors, we formalized assertions for each object or implemented additional notebook functionality to support future notebook users. Again, in all cases, we assume that the fabricator has already designed an initial 3D model or PCB layout of the artifact they wish to produce since Tandem focuses on post-modeling steps of the workflow.

7.1 Propeller

We first fabricated a drone propeller with extremely thin blades (Figure 19) that are difficult to mill (4.7mm thick, 248mm total length). Because the propeller was relatively long and thin, it was prone to heavy vibration during the bottom-up cuts. Further, we had to ensure that the line through the alignment holes was completely parallel to the machine’s y axis to avoid moving the alignment jig post-fabrication. In one instance, having both a misaligned alignment axis and insufficient tabs tore through a blade (Figure 3, 3).

To minimize the chance of tear-out occurring while milling the thin blades, we established two assertions: (1) there must be four tabs holding the propeller to the outer, two at the blade tips and two at the center of the propeller, and (2) the alignment holes must be axis aligned. To implement the first assertion, we simply ask in the notebook whether the fabricator has manually created four tabs

³On CNC mills with automatic tool changers, Florence could design an assertion by querying the machine’s hardware for the current tool. However, on more common mills without automatic tool changers, more work is required on the notebook side.

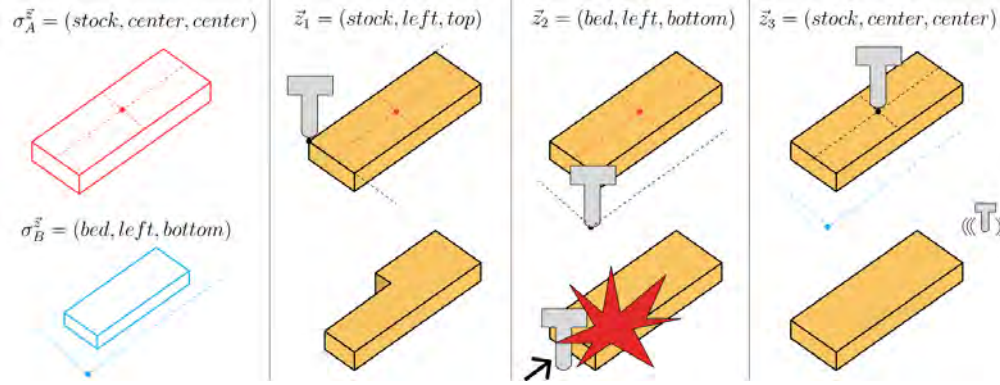


Figure 16: Potential errors resulting from differing origins between a CAM setup and a physical setup. (Left) A CAM setup σ_A^z where the origin (“zero”) is designated as the center of the stock on the x and y axes (red), and a CAM setup σ_B^z where the origin is designated as a constant location on the machine bed (blue). (Right) assuming σ_A^z , if the fabricator sets the physical zero $\tilde{z}_1$ at the top left of the stock, the mill will cut through only half of the stock. Similarly, assuming σ_A^z , with $\tilde{z}_2$ as the physical zero, the mill will crash catastrophically into the stock and the bed. Assuming σ_B^z , if the fabricator zeros on the stock at $\tilde{z}_3$, then the mill will instead cut air far from the stock.

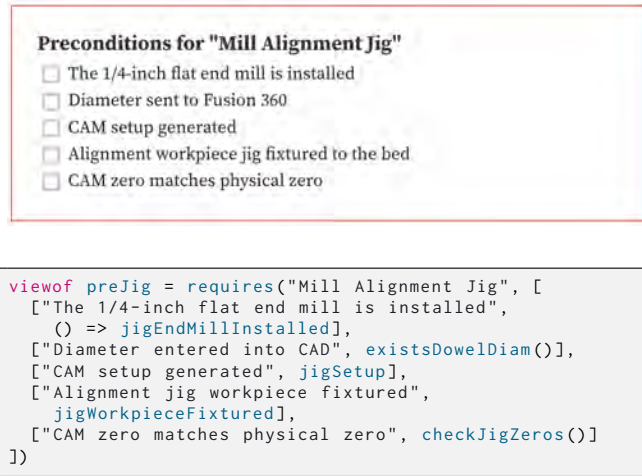


Figure 17: Preconditions for milling the alignment jig.

in CAD; to implement the second, we projected the location of the alignment holes as calculated earlier in the notebook, alongside a line representing the axis of rotation $\vec{R}$. Using a coeffect function, we manually checked whether the true alignment holes corresponded with the projected locations; if excessive deviation is observed, we added code in the notebook bridge to guide the user to re-mill the alignment holes. Finally, using the precondition function, we required both checks before a fabricator could begin the top-down and bottom-up cuts.

7.2 Spoon

Compared to the propeller, fabricating the spoon raised additional challenges due to its asymmetrical form in the XZ plane; it has a low, shallow bowl with a long, thin handle that rises up vertically

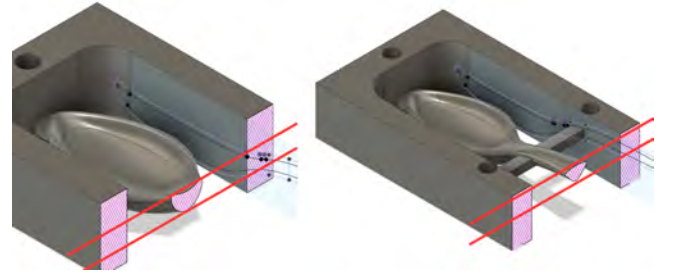


Figure 18: Minimum Thickness Assertion. From the notebook, an effect function enables Fusion 360 CAM’s section analysis tool and prompts the fabricator to inspect the minimum thicknesses of the XZ cut planes. After inspecting this, the fabricator checks a checkbox in the notebook, which returns true, fulfilling the assertion.

(see Figure 20). Despite creative placement of tabs, we found that if any part of the spoon’s handle was below a minimum thickness, the spoon bowl would tear out during milling.

To counteract this, we developed an assertion that enforces a minimum thickness requirement along the longest axis of the spoon, in our case, the y axis. Formally, we express this assertion as checking the intersection of the XZ cut plane with the spoon at all points on the y axis through the body of the spoon:

$$\forall y_{A_{min}} \leq y_i \leq y_{A_{max}} : \min_{\theta}(A \cap \text{plane}(y_i)) \geq \theta_{min},$$

where A is the body denoting the artifact, in this case the spoon, $\text{plane}(y_i)$ is the XZ cut plane with its origin at $(0, y_i, 0)$, $\min_{\theta}$ is the minimum thickness (dimension in the z direction) of the intersection of the cut plane and the artifact, and θ_{min} is a user-defined minimum thickness value.

To implement this assertion, we added code in the notebook bridge that analyzes the CAD model in Fusion 360. For now, it opens



Figure 19: The Propeller Milled from Fir.

the model in section analysis mode (Figure 18) and asks the user to manually verify that the minimum thickness is met. In the future, we aim to fully automate this process in the tool application. We added this assertion before fabricators mill anything at all; if their model is too thin, the notebook will prompt them to edit it in CAD to conform to a minimum thickness requirement.

7.3 Plywood Bowl

Next, we milled a bowl out of a glued stack of plywood. The bowl's design features multiple layers of laminated wood in its profile (Figure 21). We chose the bowl as a larger artifact to mill than the previous two, too large, in fact, to mill on most conventional mills for metal. It measures 300mm wide, 250mm long and 38mm tall.

Accordingly, we faced challenges of scale and size. First, unlike one-sided milling, the maximum thickness of any millable artifact is limited by the maximum length below the collet of the shortest end mill. To prevent fabricators from realizing this too late during the milling process and then needing to start over, we implemented additional checks in the form of a coefficient function that asks for the maximum depth-below-holder values for all end mills to be used during the process. The coefficient function then checks the height of the model and warns the user if the model is too tall for the given end mills.

Second, the main workpiece must itself be fabricated in advance from pieces of plywood cut from larger stock; we provided additional functionality for generating the CAD and CAM to mill stackable pieces to create the main stock. Namely, for a desired artifact thickness θ_a and plywood thickness θ_p , we mill exactly n pieces from the larger stock, where

$$n = \lceil \frac{\theta_a}{\theta_p} \rceil,$$

which results in the minimal necessary thickness of the workpiece that must be wasted by milling down to artifact height θ_a

$$\theta_{wasted} = n\theta_p - \theta_a.$$

7.4 Printed Circuit Board

As a final example, we implemented two-sided milling of a different type of artifact: printed circuit boards (PCBs). We implemented a separate notebook program that works with a PCB layout in Fusion 360 standalone 3D models. Our notebook program takes a 3D model that Fusion 360 generates from the fabricator's PCB layout (Figure 22) as input and generates the following CAM operations to dispatch to the mill, in order: cutting traces on the top layer, drilling through-holes, cutting traces on the bottom layer, and cutting out the outline of the entire board.

For milling PCBs, we used the Clank open-source small-format mill [58], which has a static bed and space above the tool carriage where we mounted a mini laser projector. We controlled the mill



Figure 20: A Spoon Milled from Fir.

with a Duet2 control board [9] connected directly to the fabricator's computer over a single-pair Ethernet connection. To communicate directly with the mill, we added additional functionality to the Tandem backend to forward HTTP requests sent from the notebook directly to the Duet2 control board.

8 LIMITATIONS AND FUTURE WORK

8.1 Characterizing and Formalizing Material Behavior

Currently, Tandem uses standard techniques for accommodating material constraints. For example, if a fabricator uses Tandem to generate a setup for milling a step, Tandem sets the feed and speed values of the mill to an appropriate default for the given material, such as soft wood like fir, hard wood, or plywood. Thus, some material-specific assertions are possible; for example, a notebook could prompt the user to verify that they are milling wood and not foam for fine detailing work.

However, fabricators often need to experiment with settings like feed and speed depending on the actual results that they obtain with their mill given a certain material. Tandem does not yet support automatic adjustment of parameters based on the observed response of a workpiece being milled. To address this, in future work, we plan to include support for gathering empirical data about material response. For example, for milling a propeller, Tandem could support a fabricator like Florence in helping future fabricators like Siena select an appropriate feed rate for milling the ends of the propeller blades, which are prone to excess vibration. In this case, Tandem would generate several thin test cuts, each milled with a different feed rate. Tandem would prompt Siena to select a rate that does not result in breakage yet is not excessively slow. Her selection would then fulfill an assertion that Siena must select an appropriate feed rate for milling the propeller blades.

8.2 Automatically Sensing Manual Interventions

The need for material characterization raises another opportunity for future research: integrating sensing into Tandem programs. Currently, coefficient functions rely on manual verification that a fabricator enters into the notebook by means of user interface elements. This approach offers some benefits, including prompting fabricators to physically intervene and identify the state of a physical setup at a tactile level. Further, in most cases, measuring electrical and thermal properties of a material requires manual inspection with a multimeter or thermometer, respectively. Yet, manual verification



Figure 21: A Bowl Milled from a Stack of Plywood. Because the bowl is ellipsoidal, it cannot be turned on a lathe and must be milled.

can be slow, cumbersome, and difficult to characterize features such as electrical properties or heat.

An important future line of work, then, would be integrating automated sensing into Tandem programs. One important sensor would be a camera; though we initially explored camera-projector interactions for the AR overlay, we decided to focus solely on manual interventions for the scope of this paper. In future work, we plan to leverage camera support to automate some assertions, such as finding the position and geometry of a workpiece on the machine bed. Other techniques like speckle sensing [11] or motor torque monitoring [42] could aid novices in setting reasonable initial feed and speed values like in the example above. Sensing would also let fabricators program workflow-specific versions of runtime monitoring, e.g., pausing a CNC mill if a workpiece deviates from its fixturing. We envision a larger body of work that explores implementing closed-loop control for fabrication (e.g., as explored by Piovarči et al. [56]) in Tandem.

8.3 Generalization to Non-Milling Fabrication Techniques

Though we designed and implemented Tandem as a platform for any experimental digital fabrication workflow, this paper focuses only on two-sided CNC milling, a subtractive manufacturing workflow. The single-workflow focus let us describe Tandem’s functionality in all the complexity that this single workflow entails. However, other experimental workflows, such as those mentioned in Section 2, navigate different physical-digital mismatches than those faced here. Nonetheless, this limitation also highlights a strength of Tandem: fabricators can program their own workflows and tests that are specific to additive manufacturing and laser cutting.

As an example, we briefly discuss how Tandem could help a fabricator like Florence implement an experimental 3D printing technique, Wireprint, as a notebook [46]. Wireprint is a technique for 3D printing a wireframe of an object, rather than an entire solid body; the approach is difficult to replicate and has not yet seen widespread adoption⁴. The approach involves several steps: designing a geometry in CAD that is feasible with the approach, generating wireframe triangles from the geometry, and adjusting the printer’s feed, extruder, and pause parameters.

Using Tandem, Florence could write effect functions that bring the CAD model into the notebook as an STL before passing it into the toolpath generation algorithm. She would then write code, possibly using graphical elements from coeffect functions, that help

⁴While some forms of the Cura slicer have a wireframe option, the algorithm differs from that proposed by Mueller et al. and is difficult to tune within Cura alone.

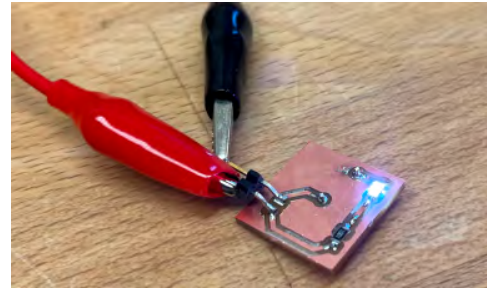


Figure 22: A Simple Two-Sided Printed Circuit Board Milled with Tandem Using a Modified Notebook.

fabricators like Siena adjust feed, extrusion, and pauses at different points in the toolpath. Florence would design the notebook to progressively print more complex parts of a wireframe, starting with a single triangle, moving to a surface of triangle, and finally to concentric slices of the model. Florence would use assertions to ensure that Siena debugs and fabricates simpler constructions before moving on to more intractable ones. Florence could also use assertions to analyze the model’s feasibility for the Wireprint approach; for example, an assertion could reject geometries with overhangs or non-concentric parts and ask the fabricator to redesign those parts. Finally, while AR functionality would likely not be useful for the Wireprint workflow, it could be useful for novel 3D printing workflows that involve precisely printing on existing geometries [8, 28].

8.4 Notebooks versus External tools

The benefit of the notebook is that Florence can lay out her steps sequentially so that Siena can follow along in one environment without the additional cognitive load of managing many applications at once. Though the notebook uses Tandem’s API to read and write to Fusion 360, the actual state of the CAD model and CAM setups still lies outside of the notebook. Siena is free to make adjustments to CAD and CAM in Fusion 360 itself, independently of the notebook’s instructions. Still, we could imagine a future where CAD and CAM functionality were part of the notebook itself; for example, a fabricator could simply call a function to open a CAM environment within the notebook. Some tools provide basic CAD functionality for web environments [53] and even for Jupyter [30] computational notebooks [26, 61]. Unfortunately, two-sided milling called for more mature CAD and CAM functionality than these tools offer. Ideally, future versions of Fusion 360 could support embedded CAD and CAM windows inside notebook environments. Alongside Fusion 360, future fabricators could expand Tandem’s backend to interface with 3D printing-specific CAM, such as Cura, Formlabs PreForm, or Slic3r, all of which have respective APIs [1, 16, 73].

8.5 Encouraging Practice and Debugging Skills

The goal of Tandem is to perform all the major steps of a digital fabrication workflow from a single program and check that prerequisites are satisfied during the workflow. It thus complements, not replaces, the need for craft-specific knowledge and hands-on tool education. The notebook scaffolds Siena’s understanding of *why*

she must do something and how the workflow works and how to avoid errors. Future iterations of fabrication-as-reproducible notebooks could include extra steps for encouraging debugging [41] and practice [38].

9 CONCLUSION

Through sustained attention to an example experimental workflow—two-sided CNC milling—we have shown why reproducing workflows is unexpectedly complicated yet crucial for research and practice. Through Tandem, we addressed this issue by making sharing experimental workflows more like sharing code. Tandem breaks the different parts that must occur into programmable components: changes to the design of 3D models, changes to the movements of machines, control of machines themselves, and the setup of the physical environment as necessitated by decisions in software. Tandem programs explicitly include unspoken preconditions that allow future fabricators to safely carry out and modify workflows, rather than leave this knowledge as informal lore passed down only in machine shops and makerspaces.

Overall, Tandem challenges an unsustainable status quo of a workflow as an abstract concept spread across multiple pieces of software and physical assumptions. Instead, Tandem provides the computational “glue” for concrete implementations of fabrication research. Through programming, Tandem unites disparate requirements for experimental fabrication in HCI and encourages a holistic, reproducible notation of physical-digital labor.

ACKNOWLEDGMENTS

We thank the anonymous reviewers whose insightful critique on an earlier manuscript helped us revise and produce a more cogent paper for publication. We also thank Sandy Kaplan for her incisive editing and feedback on earlier drafts of the paper. Finally, this work was graciously supported by the National Science Foundation (award #2024435) and by the Alfred P. Sloan Foundation.

REFERENCES

- [1] Alessandro Ranellucci. 2023. Slic3r - Open source 3D printing toolbox. <https://slic3r.org/>
- [2] Fabien C. Y. Benureau and Nicolas P. Rougier. 2018. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. *Frontiers in Neuroinformatics* 11 (2018). <https://www.frontiersin.org/articles/10.3389/fninf.2017.00069>
- [3] Bray Lab Contributors. 2023. Flip-Cut (2 Sided Milling) Instructions - Bray Lab. <https://sites.tufts.edu/bray/theshop/cnc-router/flip-cut-2-sided-milling-instructions/>
- [4] Leah Buechley and Ruby Ta. 2023. 3D Printable Play-Dough: New Biodegradable Materials and Creative Possibilities for Digital Fabrication. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3544548.3580813>
- [5] M. Burnett, C. Cook, O. Pendse, G. Rothermel, J. Summet, and C. Wallace. 2003. End-user software engineering with assertions in the spreadsheet paradigm. In *25th International Conference on Software Engineering, 2003. Proceedings*. 93–103. <https://doi.org/10.1109/ICSE.2003.1201191> ISSN: 0270-5257.
- [6] Dan Cascaval, Mira Shalah, Phillip Quinn, Rastislav Bodik, Maneesh Agrawala, and Adriana Schulz. 2021. Differentiable 3D CAD Programs for Bidirectional Editing. *arXiv:2110.01182 [cs]* (Oct. 2021). <http://arxiv.org/abs/2110.01182> arXiv: 2110.01182.
- [7] Ishan Chatterjee, Tadeusz Pforte, Aspen Tng, Farshid Salemi Parizi, Chaoran Chen, and Shwetak Patel. 2022. ARDW: An Augmented Reality Workbench for Printed Circuit Board Debugging. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–16. <https://doi.org/10.1145/3526113.3545684>
- [8] Xiang ‘Anthony’ Chen, Jeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 29–39. <https://doi.org/10.1145/2984511.2984512>
- [9] David Crocker. 2023. Duet 3D. <https://www.duet3d.com/>
- [10] Audrey Desjardins and Timea Tihanyi. 2019. ListeningCups: A Case of Data Tacitly and Data Stories. In *Proceedings of the 2019 on Designing Interactive Systems Conference (DIS '19)*. Association for Computing Machinery, New York, NY, USA, 147–160. <https://doi.org/10.1145/3322276.3323694>
- [11] Mustafa Doga Dogan, Steven Vidal Acevedo Colon, Varnika Sinha, Kaan Aksit, and Stefanie Mueller. 2021. SensiCut: Material-Aware Laser Cutting Using Speckle Sensing and Deep Learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 24–38. <https://doi.org/10.1145/3472749.3474733>
- [12] Eythor R. Eiríksón, David B. Pedersen, Jeppe R. Frisvad, Linda Skovmand, Valentin Heun, Pattie Maes, and Henrik Aanæs. 2017. Augmented Reality Interfaces for Additive Manufacturing. In *Image Analysis*, Puneet Sharma and Filippo Maria Bianchi (Eds.), Vol. 10269. Springer International Publishing, Cham, 515–525. https://doi.org/10.1007/978-3-319-59126-1_43 Book Title: Image Analysis Series Title: Lecture Notes in Computer Science.
- [13] Steven Feiner, Blair Macintyre, and Dorée Seligmann. 1993. Knowledge-based augmented reality. *Commun. ACM* 36, 7 (July 1993), 53–62. <https://doi.org/10.1145/159544.159587>
- [14] Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: Remixing Physical Objects with Copy and Paste from the Real World. In *Adjunct Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 381–382. <https://doi.org/10.1145/1866218.1866230>
- [15] Jack Forman, Mustafa Doga Dogan, Hamilton Forsythe, and Hiroshi Ishii. 2020. DefeXtiles: 3D Printing Quasi-Woven Fabric via Under-Extrusion. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1222–1233. <https://doi.org/10.1145/3379337.3415876>
- [16] Formlabs. 2023. PreForm. <https://formlabs.com/software/#preform>
- [17] Frick Fossdal, Rogardt Haldal, and Nadya Peek. 2021. Interactive Digital Fabrication Machine Control Directly Within a CAD Environment. In *Symposium on Computational Fabrication (SCF '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3485114.3485120>
- [18] Frick H Fossdal, Vinh Nguyen, Rogardt Haldal, Corie L. Cobb, and Nadya Peek. 2023. Vespidae: A Programming Framework for Developing Digital Fabrication Workflows. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference (DIS '23)*. Association for Computing Machinery, New York, NY, USA, 2034–2049. <https://doi.org/10.1145/3563657.3596106>
- [19] Andrew Gleadall. 2021. FullControl GCode Designer: Open-source software for unconstrained design in additive manufacturing. *Additive Manufacturing* 46 (Oct. 2021), 102109. <https://doi.org/10.1016/j.addma.2021.102109>
- [20] Ammar Hattab and Gabriel Taubin. 2019. Interactive Fabrication of CSG Models with Assisted Carving. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. Association for Computing Machinery, New York, NY, USA, 677–682. <https://doi.org/10.1145/3294109.3295644>
- [21] Ammar Hattab and Gabriel Taubin. 2019. Rough carving of 3D models with spatial augmented reality. In *Proceedings of the ACM Symposium on Computational Fabrication (SCF '19)*. Association for Computing Machinery, Pittsburgh, Pennsylvania, 1–10. <https://doi.org/10.1145/3328939.3328998>
- [22] Ian Holland and Jamie A. Davies. 2020. Automation in the Life Science Research Laboratory. *Frontiers in Bioengineering and Biotechnology* 8 (2020). <https://www.frontiersin.org/article/10.3389/fbioe.2020.571777>
- [23] Gaoping Huang, Xun Qian, Tianyi Wang, Fagun Patel, Maitreya Sreeram, Yuanzhi Cao, Karthik Ramani, and Alexander J. Quinn. 2021. AdapTutAR: An Adaptive Tutoring System for Machine Tasks in Augmented Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3411764.3445283>
- [24] Jennifer Jacobs and Leah Buechley. 2013. Codeable Objects: Computational Design and Digital Fabrication for Novice Programmers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1589–1598. <https://doi.org/10.1145/2470654.2466211>
- [25] Jeremiah Langston. 2020. Milling a 2 side contoured part using dowel pin indexing features (Fusion 360). <https://www.youtube.com/watch?v=nh5sYFraaBI>
- [26] JupyterCAD Development Team. 2023. JupyterCAD — JupyterCAD documentation. <https://jupytercad.readthedocs.io/en/latest/>
- [27] Jun Kato and Masataka Goto. 2017. f3.js: A Parametric Design Tool for Physical Computing Devices for Both Interaction Designers and End-users. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. Association for Computing Machinery, New York, NY, USA, 1099–1110. <https://doi.org/10.1145/3064663.3064681>

- [28] Jeeun Kim, Clement Zheng, Haruki Takahashi, Mark D Gross, Daniel Ashbrook, and Tom Yeh. 2018. Compositional 3D Printing: Expanding & Supporting Workflows Towards Continuous Fabrication. In *Proceedings of the 2Nd ACM Symposium on Computational Fabrication (SCF '18)*. ACM, New York, NY, USA, 5:1–5:10. <https://doi.org/10.1145/3213512.3213518>
- [29] Yoonji Kim, Hyein Lee, Ramkrishna Prasad, Seungwoo Je, Youngkyung Choi, Daniel Ashbrook, Ian Oakley, and Andrea Bianchi. 2020. SchemaBoard: Supporting Correct Assembly of Schematic Circuits using Dynamic In-Situ Visualization. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 987–998. <https://doi.org/10.1145/3379337.3415887>
- [30] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Proceedings of the 20th International Conference on Electronic Publishing*. Fernando Loizides and Birgit Schmidt (Eds.). IOS Press, 87–90. <https://doi.org/10.3233/978-1-61499-649-1-87>
- [31] Marion Koelle, Madalina Nicolae, Aditya Shekhar Nittala, Marc Teyssier, and Jürgen Steimle. 2022. Prototyping Soft Devices with Interactive Bioplastics. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, 1–16. <https://doi.org/10.1145/3526113.3545623>
- [32] Hadas Kress-Gazit, Kerstin Eder, Guy Hoffman, Henny Admoni, Brenna Argall, Rüdiger Ehlers, Christoffer Heckman, Nils Jansen, Ross Knepper, Jan Křetínský, Shelly Levy-Tzedek, Jamy Li, Todd Murphey, Laurel Riek, and Dorsa Sadigh. 2021. Formalizing and guaranteeing human-robot interaction. *Commun. ACM* 64, 9 (Aug. 2021), 78–84. <https://doi.org/10.1145/3433637>
- [33] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. 2018. Synthesis for Robots: Guarantees and Feedback for Robot Behavior. *Annu. Rev. Control Robot. Auton. Syst.* 1, 1 (May 2018), 211–236. <https://doi.org/10.1146/annurev-control-060117-104838>
- [34] Don LaCourse. 2017. 2-Sided (Flip) Machining Explored. <https://mecsoft.com/blog/2-sided-flip-machining-explored/>
- [35] Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. Tsugite: Interactive Design and Fabrication of Wood Joints. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 317–327. <https://doi.org/10.1145/3379337.3415899>
- [36] Lauren Lee McCarthy. 2023. p5.js. <https://p5js.org/>
- [37] Mackenzie Leake, Gilbert Bernstein, Abe Davis, and Maneesh Agrawala. 2021. A mathematical foundation for foundation paper pieceable quilts. *ACM Trans. Graph.* 40, 4 (July 2021), 65:1–65:14. <https://doi.org/10.1145/3450626.3459853>
- [38] Mackenzie Leake, Kathryn Jin, Abe Davis, and Stefanie Mueller. 2023. InStitches: Augmenting Sewing Patterns with Personalized Material-Efficient Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3544548.3581499>
- [39] Jingyi Li, Jennifer Jacobs, Michelle Chang, and Björn Hartmann. 2017. Direct and Immediate Drawing with CNC Machines. In *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication (SCF '17)*. ACM, New York, NY, USA, 11:1–11:2. <https://doi.org/10.1145/3083157.3096344>
- [40] Min Liu, Yunbo Zhang, Jing Bai, Yuanzhi Cao, Jeffrey M. Alperovich, and Karthik Raman. 2017. WireFab: Mix-Dimensional Modeling and Fabrication for 3D Mesh Models. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 965–976. <https://doi.org/10.1145/3025453.3025619>
- [41] Joanne Lo, Cesar Torres, Isabel Yang, Jasper O'Leary, Danny Kaufman, Wilnot Li, Mira Dontcheva, and Eric Paulos. 2016. Aesthetic Electronics: Designing, Sketching, and Fabricating Circuits Through Digital Exploration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 665–676. <https://doi.org/10.1145/2984511.2984579>
- [42] Benjamin Lutz, Philip Howell, Daniel Regulin, Bastian Engelmann, and Jörg Franke. 2021. Towards Material-Batch-Aware Tool Condition Monitoring. *Journal of Manufacturing and Materials Processing* 5, 4 (Dec. 2021), 103. <https://doi.org/10.3390/jmmp5040103> Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
- [43] Chandan Mahapatra, Jonas Kjeldmand Jensen, Michael McQuaid, and Daniel Ashbrook. 2019. Barriers to End-User Designers of Augmented Fabrication. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, Glasgow, Scotland Uk, 1–15. <https://doi.org/10.1145/3290605.3300613>
- [44] Martin Barfoed. 2016. Two Sided 3D Milling using Pin Method. CAD and CAM in Fusion 360. <https://www.youtube.com/watch?v=Ce6mYKMAoX8>
- [45] McNeel. 2023. Grasshopper. <https://www.grasshopper3d.com/>
- [46] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D printed pre-views for fast prototyping. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 273–280. <https://doi.org/10.1145/2642918.2647359>
- [47] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. *Proceedings of the 25th annual ACM symposium on User interface software and technology* (Oct. 2012), 599–606. <https://doi.org/10.1145/2380116.2380191> Conference Name: UIST '12: The 25th Annual ACM Symposium on User Interface Software and Technology ISBN: 9781450315807 Place: Cambridge Massachusetts USA Publisher: ACM.
- [48] Florian Müller, Maximilian Barnikol, Markus Funk, Martin Schmitz, and Max Mühlhäuser. 2018. CaMea: Camera-Supported Workpiece Measurement for CNC Milling Machines. In *Proceedings of the 11th Pervasive Technologies Related to Assistive Environments Conference (PETRA '18)*. Association for Computing Machinery, New York, NY, USA, 345–350. <https://doi.org/10.1145/3197768.3201569>
- [49] Chandrakana Nandi, James R. Wilcox, Pavel Panchekha, Taylor Blau, Dan Grossman, and Zachary Tatlock. 2018. Functional Programming for Compiling and Decompiling Computer-aided Design. *Proc. ACM Program. Lang.* 2, ICFP (July 2018), 99:1–99:31. <https://doi.org/10.1145/3236794>
- [50] National Academies. 2019. *Reproducibility and Replicability in Science*. National Academies Press.
- [51] Observable Notebook. 2023. Observable Notebook. <https://observablehq.com>
- [52] Alex Olwal, Jonny Gustafsson, and Christoffer Lindfors. 2008. Spatial augmented reality on industrial CNC-machines. In *Proceedings of Electronic Imaging 2008*, Ian E. McDowall and Margaret Dolinsky (Eds.). San Jose, CA, 680409. <https://doi.org/10.1117/12.760960>
- [53] Onshape. 2022. Onshape. <https://www.onshape.com/en/>
- [54] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 579:1–579:12. <https://doi.org/10.1145/3173574.3174153>
- [55] Jeffrey M. Perkel. 2021. Reactive, reproducible, collaborative: computational notebooks evolve. *Nature* 593, 7857 (May 2021), 156–157. <https://doi.org/10.1038/d41586-021-01174-w>
- [56] Michal Piovarči, Michael Foshey, Jie Xu, Timothy Erps, Vahid Babaei, Piotr Didyk, Szymon Rusinkiewicz, Wojciech Matusik, and Bernd Bickel. 2022. Closed-loop control of direct ink writing via reinforcement learning. *ACM Trans. Graph.* 41, 4 (July 2022), 112:1–112:10. <https://doi.org/10.1145/3528223.3530144>
- [57] Troels A Rasmussen and Timothy Merritt. 2018. ProjecTables: Augmented CNC tools for sustainable creative practices. *International Journal of Architectural Computing* 16, 3 (Sept. 2018), 227–242. <https://doi.org/10.1177/1478077118792356> Publisher: SAGE Publications.
- [58] Jake Read. 2023. Clank: Fabricatable, Modular CNC. <https://clank.tools/>
- [59] Nico Ritschel, Anand Ashok Sawant, David Weintrop, Reid Holmes, Alberto Bacchelli, Ronald Garcia, Chandrika K R, Avijit Mandal, Patrick Francis, and David C. Shepherd. 2023. Training industrial end-user programmers with interactive tutorials. *Software: Practice and Experience* 53, 3 (2023), 729–747. <https://doi.org/10.1002/spe.3167> _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.3167>
- [60] Matthew Ryan. 2022. Immersive Virtual Reality Error Management Training for CNC Machining Setup Procedures. *Theses* (April 2022). <https://scholarworks.rit.edu/theses/11190>
- [61] Phil Schatzmann. 2023. OpenSCAD Kernel for Jupyter. <https://github.com/pschatzmann/jupyter-openscad-kernel> original-date: 2020-02-26T09:42:36Z.
- [62] Erik Seligman, E. Thomas Schubert, and M. V. Achutha Kiran Kumar. 2015. *Formal verification: an essential toolkit for modern VLSI design*. Elsevier/MK, Morgan Kaufmann is an imprint of Elsevier, Amsterdam ; Boston.
- [63] Shopbot Tools. 2022. ShopBot. <https://www.shopbottools.com/products/accessories/atc>
- [64] Thomas Stevens. 2019. Rotation error when flipping stock - CNC Machines / Shapeoko. <https://community.carbide3d.com/t/rotation-error-when-flipping-stock/15236/6>
- [65] Blair Subbaraman and Nadya Peek. 2022. p5.fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference (DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1148–1161. <https://doi.org/10.1145/3532106.3533496>
- [66] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented Reality and Robotics: A Survey and Taxonomy for AR-enhanced Human-Robot Interaction and Robotic Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–33. <https://doi.org/10.1145/3491102.3517719>
- [67] Sławomir K. Tadeja, Luca O. Solari Bozzi, Kerr D. G. Samson, Sebastian W. Patinson, and Thomas Bohné. 2023. Exploring the repair process of a 3D printer using augmented reality-based guidance. *Computers & Graphics* 117 (Dec. 2023), 134–144. <https://doi.org/10.1016/j.cag.2023.10.017>
- [68] Rundong Tian and Eric Paulos. 2021. Adroid: Augmenting Hands-on Making with a Collaborative Robot. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 270–281. <https://doi.org/10.1145/3472749.3474749>

- [69] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, NY, USA, 713–725. <https://doi.org/10.1145/3332165.3347918>
- [70] Iremnur Tokac, J. Philips, H. Bruyninckx, and A. Vande Moere. 2021. Fabrication grammars: bridging design and robotics to control emergent material expressions. *Construction Robotics* (2021). <https://link.springer.com/article/10.1007/s41693-021-00053-0>
- [71] Jasper Tran O'Leary, Gabrielle Benabdallah, and Nadya Peek. 2023. Imprimer: Computational Notebooks for CNC Milling. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3544548.3581334>
- [72] Jasper Tran O'Leary, Eunice Jun, and Nadya Peek. 2022. Improving Programming for Exploratory Digital Fabrication with Inline Machine Control and Styled Toolpath Visualizations. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication (SCF '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3559400.3561998>
- [73] Ultimaker. 2021. Ultimaker/Cura Printer Definitions. <https://github.com/Ultimaker/Cura>
- [74] Vectric Ltd. 2021. Introduction to Two Sided Machining | Vectric V11 Tutorials. <https://www.youtube.com/watch?v=wfaqg7m9erw>
- [75] Justin Vrana, Orlando de Lange, Yaoyu Yang, Garrett Newman, Ayesha Saleem, Abraham Miller, Cameron Cordray, Samer Halabiya, Michelle Parks, Eriberto Lopez, Sarah Goldberg, Benjamin Keller, Devin Strickland, and Eric Klavins. 2021. Aquarium: open-source laboratory software for design, execution and data management. *Synthetic Biology* 6, 1 (Jan. 2021). <https://doi.org/10.1093/synbio/ysab006>
- [76] 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 (UIST '15)*. ACM, New York, NY, USA, 93–102. <https://doi.org/10.1145/2807442.2807451>
- [77] Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2020. Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Reconstruction. *arXiv:2010.02392 [cs]* (Oct. 2020). <http://arxiv.org/abs/2010.02392> arXiv: 2010.02392.
- [78] Max Willsey, Ashley P. Stephenson, Chris Takahashi, Pranav Vaid, Bichlien H. Nguyen, Michal Piszczek, Christine Betts, Sharon Newman, Sarang Joshi, Karin Strauss, and Luis Ceze. 2019. Puddle: A Dynamic, Error-Correcting, Full-Stack Microfluidics Platform. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (AS-PLOS '19)*. Association for Computing Machinery, Providence, RI, USA, 183–197. <https://doi.org/10.1145/3297858.3304027>
- [79] Chenming Wu, Haisen Zhao, Chandrakana Nandi, Jeffrey I. Lipton, Zachary Tatlock, and Adriana Schulz. 2019. Carpentry Compiler. *ACM Trans. Graph.* 38, 6 (Nov. 2019), 195:1–195:14. <https://doi.org/10.1145/3355089.3356518>
- [80] Junichi Yamaoka and Yasuaki Kakehi. 2016. MiragePrinter: interactive fabrication on a 3D printer with a mid-air display. In *ACM SIGGRAPH 2016 Studio (SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, 1–2. <https://doi.org/10.1145/2929484.2929489>
- [81] Chih-Kai Yang, Yu-Hsi Chen, Tung-Jui Chuang, Kalpana Shankhwar, and Shana Smith. 2020. An augmented reality-based training system with a natural user interface for manual milling operations. *Virtual Reality* 24, 3 (Sept. 2020), 527–539. <https://doi.org/10.1007/s10055-019-00415-8>
- [82] Humphrey Yang, Danli Luo, Kuanren Qian, and Lining Yao. 2021. Freeform Fabrication of Fluidic Edible Materials. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3411764.3445097>
- [83] Hironori Yoshida, Takeo Igarashi, Yusuke Obuchi, Yosuke Takami, Jun Sato, Mika Araki, Masaaki Miki, Kosuke Nagata, Kazuhide Sakai, and Syunsuke Igarashi. 2015. Architecture-scale human-assisted additive manufacturing. *ACM Trans. Graph.* 34, 4 (July 2015), 88:1–88:8. <https://doi.org/10.1145/2766951>
- [84] Clement Zheng, Bo Han, Xin Liu, Laura Devendorf, Hans Tan, and Ching Chiuan Yen. 2023. Crafting Interactive Circuits on Glazed Ceramic Ware. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. <https://doi.org/10.1145/3544548.3580836>
- [85] Clement Zheng, Zhen Zhou Yong, Hongnan Lin, HyunJoo Oh, and Ching Chiuan Yen. 2022. Shape-Haptics: Planar & Passive Force Feedback Mechanisms for Physical Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3491102.3501829>
- [86] 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. <https://doi.org/10.1145/2470654.2481361>