



Vespidae: A Programming Framework for Developing Digital Fabrication Workflows

Frikk H Fossdal
ffo@hvl.no
Western Norway University of
Applied Sciences
Bergen, Norway

Vinh Nguyen
vinhqn@uw.edu
University of Washington,
Integrated Fabrication Lab
Seattle, Washington, USA

Rogardt Heldal
rohe@hvl.no
Western Norway University of
Applied Sciences
Bergen, Norway

Corie L. Cobb
clcobb@uw.edu
University of Washington,
Integrated Fabrication Lab
Seattle, Washington, USA

Nadya Peek
nadya@uw.edu
University of Washington,
Machine Agency
Seattle, Washington, USA

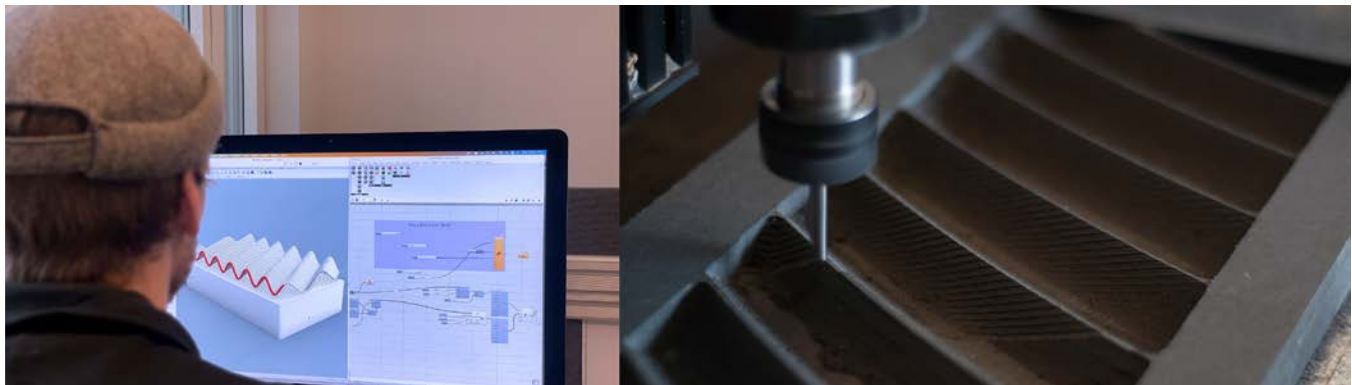


Figure 1: Vespidae is a programming framework to support the development of custom tool paths and visualizations for digital fabrication. Software modules provide the computational functions necessary to generate toolpaths (red), and transform these into executable machine code. This enables higher levels of control over toolpathing operations while still harnessing the benefits of abstraction; shown here is an undulating toolpath used to texture a piece of wood using a CNC milling machine.

ABSTRACT

Digital fabrication machines are controlled through code. Software that generates this code, such as slicers, often rely on abstractions that restrict practitioners from exploring the full design space. We contribute Vespidae, a programming framework for developing custom toolpaths and visualizations. Vespidae module types include Toolpaths, Actions, Solvers, and Export. These generate geometry, specify machine tasks, sort and visualize action sequences, and generate and stream machine code. We show example workflows that demonstrate Vespidae's strengths in supporting iteration and unconventional practice. These include non-planar 3D printing, varying a print's tactile qualities with under-extrusion, and exploring the design space of milling marks. Furthermore, we used

Vespidae over the course of six months to explore multi-material 3D printing for energy storage devices on a custom machine. Finally, we discuss how Vespidae contributes to a movement in HCI arguing for human-machine collaboration.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools.**

KEYWORDS

Digital Fabrication; Machine Control; CAD/CAM; Digital Craftsmanship; 3D Printing; Energy Storage Devices



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

DIS '23, July 10–14, 2023, Pittsburgh, PA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9893-0/23/07.
<https://doi.org/10.1145/3563657.3596106>

ACM Reference Format:

Frikk H Fossdal, Vinh Nguyen, Rogardt Heldal, Corie L. Cobb, and Nadya Peek. 2023. Vespidae: A Programming Framework for Developing Digital Fabrication Workflows. In *Designing Interactive Systems Conference (DIS '23)*, July 10–14, 2023, Pittsburgh, PA, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3563657.3596106>

1 INTRODUCTION

A laser cutter has X, Y, and Z axes—the height of the head can be raised or lowered to accommodate different stock materials and the XY can cut out any 2D shape. There is no mechanical reason that a laser cutter could not vary its Z-height while cutting in XY. Yet laser cutter software is not designed to accommodate this type of toolpath. Other digital fabrication toolpath generation software similarly uses a 2D+1D approach. 3D printing slicers default to planar layer-by-layer printing. Computer-controlled milling cuts one ‘pass-depth’ at a time. These types of abstractions—abstractions that are built into toolpath generation software—make assumptions about how we engage with material, machine, and process. These abstractions can reduce risk—machine crashes, material failures—but they also restrict us from exploring the possibilities at the intersection of material and machine.

Andersen et al. [3] push against machines as rote executors of automation: “A new machine is full of open questions, if we approach it well. With no rigid social hierarchy and norms already set around a new technology, we are able to push against its constraints.” To collaborate with machines, they argue for fluidity with their control: “digital crafts-machine-ship is about encoding the rules of making, not to preserve them but so that they may be more easily modified and changed.” Inspired by this and rich prior work in HCI [12, 23, 37, 65], we seek to support further exploration of the digital crafts-machine-ship design space.

Pushing against the constraints of fabrication machines and systems, HCI researchers have developed many novel digital fabrication processes, for example, for fabricating microstructure [18, 38, 47], electronic devices [34], biodegradable electronic devices [58, 68], textile devices [1, 27, 49], and shape-shifting devices [71, 73]. Many of these novel workflows rely on modifying a key step in the fabrication process, specifically by authoring custom machine instructions. In established fabrication processes, these machine instructions are typically generated in CAM (computer-aided manufacturing), software that encapsulates the domain expertise and best practices of a fabrication process into a more user-friendly and understandable interface. For example, FDM (fused deposition modeling) 3D printing CAM software takes a 3D model (often in .stl format) and a set of user parameters (e.g., support settings, quality) to produce a sequence of toolpaths (often as a .gcode file). This software is also called a ‘slicer’, emphasizing the planar toolpaths they typically generate. The slicer abstracts away the complex geometrical computation it conducts to generate machine instructions, and provides simple visualizations of the resulting toolpaths. However, it does not support the unconventional toolpath authoring needed for many of the processes above, nor does it support simulating or visualizing physical outcomes of unusual toolpaths. Practitioners who wish to explore this need to develop custom workarounds. We refer to these kinds of workflows as unconventional digital fabrication.

For example, for *ListeningCups*, Desjardins and Tihanyi [10] developed a simple workflow to embed audio data into G-code by manually inserting lines of code that paused the printer. In these pauses, the porcelain continued oozing out of the nozzle, generating characteristic bumps on the cup surface. To get their preferred

texture, they “played” with the machine, exploring how simple modifications led to different material outcomes. They describe needing to rely on “*anticipated tactility*”, noting, “*This close relationship with the machine was possible because Tihanyi already had an excellent understanding and anticipation of what the printer could do, including the innovative use of the dwell command for the production of texture.*” We ask how this type of exploration can be better supported for more machines, materials, and processes, allowing us to venture towards less-simple instructions or less-familiar human-machine relationships.

Towards this goal, we identify a bottleneck at creating and visualizing machine instructions that are unconventional. Therefore, our aim is to support the creation of novel machine instruction authoring and visualization tools. This way, practitioners in digital fabrication who wish to explore the boundaries of their machines and materials can easily iterate and experiment with instructions. We present *Vespidae*, a programming framework for digital fabrication toolpaths. Here we consider a programming framework tool that provides ready-made components that can be customized, including libraries, APIs, and compilers. We designed *Vespidae* to expose key computational steps of going from geometric parameters to machine code in individual modules. Specifically, we contribute software modules for creating, manipulating, and visualizing toolpaths; software modules for machine-specific translation (e.g., G-code flavours); and software modules for uploading programs and communicating with machine controllers. *Vespidae* is named after the insect family that contains eusocial wasps, which are highly cooperative creatures. *Vespidae* is implemented as an open-source plugin for Rhino/Grasshopper, and its insect-related naming is part of an informal convention.

2 RELATED WORK

In this section, we describe related research efforts and what distinguishes our approach.

HCI researchers have developed numerous fabrication methods that rely on custom digital fabrication machine instructions. For example, *DefeXtiles* uses periodic under-extrusion in 3D printing to create flexible devices using a rigid material [18]. Others use bridging [62, 63], springs [25], and foam-inspired structure [67] to introduce flexibility in rigid devices. Novel shape-shifting and morphing structures can also be created by controlling 3D printing parameters [70, 71]. Researchers have tuned material properties by printing patterned microstructure to introduce metamaterial behavior [31] and have also modified machine instructions as a carrier for other forms of data, such as audio [10] and information about the objects themselves [13]. Developing these fabrication methods requires low-level control and iteration [65]. We seek to support this type of inquiry by assisting practitioners in authoring unconventional machine instructions.

In addition to unconventional fabrication methods, HCI researchers are also exploring unconventional materials. Sustainability is a priority in HCI, and has led to the development of biodegradable materials for use in digital fabrication [39, 58, 68]. HCI researchers have also developed other novel materials, for example with shape-changing or sensing properties [51]. Digital fabrication methods need to be tuned and optimized for each of

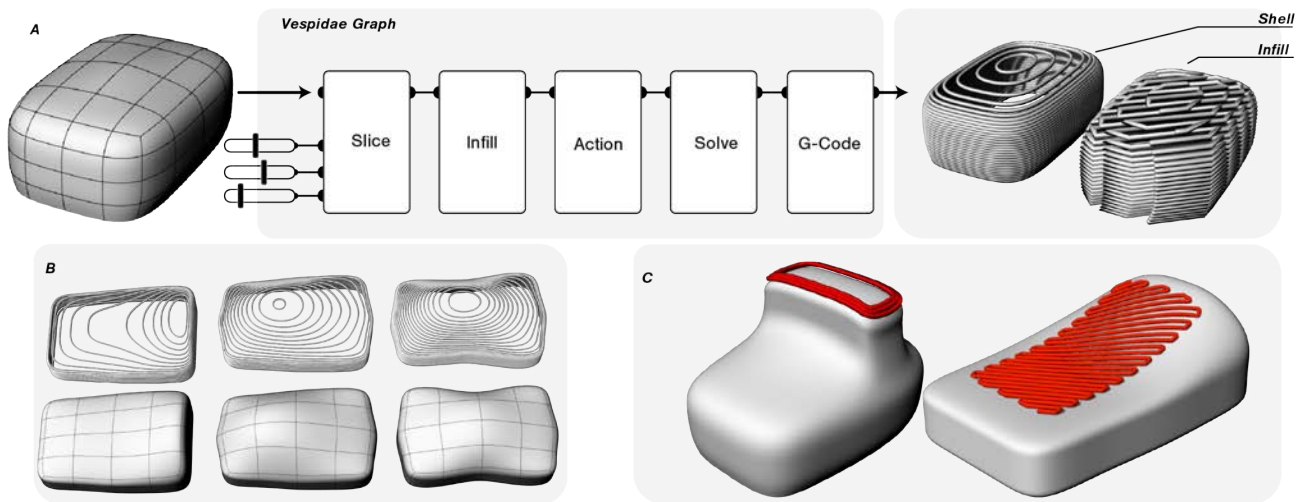


Figure 2: Arne's Vespidae program. A: At a high level, Arne's program represents a traditional slicer. However, each module exposes parameters that invite modification and change. Changes are immediately visualized. B: These visualizations allow Arne to quickly understand how his design (below) may translate when sliced (above). C: Vespidae enables the creation of custom 3D geometry, such as printing on top of existing objects/geometry.

these novel materials. Vespidae supports these efforts by enabling rapid iteration to efficiently optimize manufacturing parameters.

Finally, HCI researchers often build custom digital fabrication machines that contribute novel functionality and interaction paradigms. These new machines frequently add novel end-effectors to existing machines, examples of which include end-effectors that print with woolen yarn [27], gooey soybeans [72], and human-safe hydrogels [54]. However, even simple machine modifications with novel end-effectors require custom machine control code. This is achieved by repurposing existing infrastructure, which often involves hacking slicers meant for off-the-shelf 3D printers. However, this approach can be both limiting and challenging. In describing *“the pliable machine”*, Landwehr Sydow et al. [37] emphasize the need for an “accessible” machine which is open to modification and change. Lowering the barrier to making open and customizable machines represents another approach [20, 48, 69], but this line of inquiry focuses mainly on hardware and less on the software for design and control. We seek to support people who are building new hardware, whether through hacking or from scratch, by providing a framework for developing control software. In the spirit of pliable machines, which concerns hardware, software, and social context, *“where makers are constantly re-configuring the machine in terms of its boundaries and limitations”* [37], we seek to support a practice that Somanath et al. [57] describe where *“tools and creative practices can be reconfigured in ways to explore the full range of ways these technologies affect the creative process.”*

These novel and exploratory fabrication methods, experimental materials, and custom (pliable) machines demonstrate a clear need for tools that support authoring machine instructions. Researchers have highlighted the expertise that comes from understanding the interplay between machine and instruction [3, 36, 74], however, systems research exploring this has been limited. P5.fab is a system that lets users program 3D printers directly using creative coding

language p5.js [60]. However, p5.fab provides limited scaffolding for the machine programming. In contrast, Vespidae provides modules that support toolpath design directly. FullControl GCode Designer lets users specify toolpaths directly for additive manufacturing [21], but they use the program Excel as their front end, which does not support complex geometry or geometric operations. Most similar to Vespidae, Fossdal et al. [19] contribute software for interfacing with digital fabrication machines directly from a CAD environment. However, they demonstrate 2D toolpaths. Here, we showcase fully 3D toolpath control for multi-material printing.

Fabrication research in HCI has contributed many systems that lower the barrier to entry [46]. Many of these systems support novices taking on fabrication tasks [2, 26]. Using Vespidae relies on expertise in digital design, and is not targeted specifically at a fabrication novice. Rather, Vespidae is for users who are limited by currently available tools and the inability to easily author machine instructions for them. We argue that developing fabrication systems for domain experts also leads to insights that can be used when developing systems for other groups of users, including novices. We draw from prior HCI research in toolkits and creativity support tools to guide us in our system implementation and evaluation [40, 52], and describe implications for design of future systems in Section 7.

3 WALK THROUGH: USING VESPIDAE TO CREATE A SLICER

In this section, we describe how Vespidae can be used to create a custom slicing program for a FDM 3D printer. This is a simple example of how someone might use Vespidae modules.

To begin our walkthrough, Arne is a designer developing a product that will be 3D printed. He wants to explore trade-offs between possible input geometries and specific print outcomes. Using an

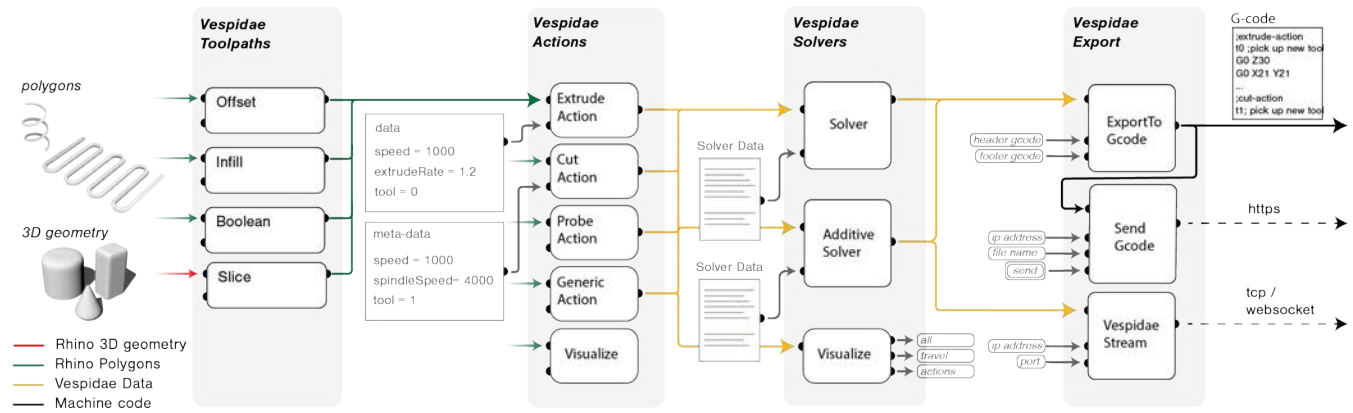


Figure 3: An overview of all the Vespidae modules and the dataflow between them. Toolpathing programs are made by combining modules together in graphs. Users are free to mix and match modules from the different categories as long as they adhere to the correct input and output data format, e.g., curves, Actions, G-code.

off-the-shelf slicer doesn't let him control the grain of the print in the ways he wants to. Therefore, he decides to create his own Vespidae program to explore the design space.

First, he downloads and imports the Vespidae library into the CAD software Rhino/Grasshopper. He creates a program by dragging modules (also known in Grasshopper as 'components') onto his Grasshopper canvas and connecting them. He also designs an initial soap bar 3D geometry in Rhino. He uses the Vespidae Slice-module to create 2D slices, or polygons, from its 3D geometry. He then uses the Vespidae Infill-module to generate a fill pattern inside each of the sliced polygons. Together, this geometry creates his initial **toolpath**. He then takes the data and processes it, tagging specific geometries with **Actions** that describe the fabrication process. This allows him to describe process-specific parameters like extrusion rate, temperature, and speed. He then groups the actions using a Vespidae Solve-module. Vespidae **Solvers** convert sequences of Actions into complete programs by sorting the execution order of toolpaths and generating travel moves between them. Arne is particularly excited about the possibilities here, as he knows that minimizing travel moves and sorting into as-continuous-as-possible toolpaths will lead to better finish on his final print. Finally, he uses a Vespidae **Export**-module to generate G-code specific to his machine. Arne's program is shown in Figure 2.

Although from a high-level this workflow seems like a traditional slicer, it differs in that its entire computational stack is exposed and editable. This means that Arne is free to tweak and customize each step. For example, if he wanted to replace the infill with a manually drawn pattern, he can easily do so with the Vespidae Infill-module. Alternatively, if Arne wanted to have the machine first print a manually drawn toolpath, he could assign this to an Action that is placed first by the Solver. In this workflow, the execution order of Actions can easily be reorganized, and custom Actions can easily be defined.

Because the Vespidae program runs in his 3D modelling program, Arne can easily visualize many of the fabrication steps and design them with more complex geometric input. For example, he can

quickly get a sense of how the scale of his soap bar compares to the filament thickness in the print. He can also see how different geometries may translate when sliced (see Figure 2B). Instead of using the Vespidae Slicer module, he can also use his input geometry to create non-planar toolpaths, enabling him to fine-tune his outcomes (see the red non-planar toolpaths in Figure 2C).

At last, Arne saves his Vespidae program as a .gh Grasshopper file. He loads G-code he generated with his program onto a memory card and brings it to his office to print. Arne also has the flexibility of uploading the code to his printer directly over USB. In this way, he could have iterated not just with visualizations, but also with tangible printer outcomes. We describe the Vespidae Toolpath, Action, Solver, and Export modules in more detail in the next section.

4 VESPIDAE SYSTEM OVERVIEW

The core philosophy for Vespidae is to enable a more playful way of both interacting with and extending the functionality of digital fabrication machines. The implementation of Vespidae is the result of the authors practicing digital fabrication themselves, both as design practitioners using the machines to make objects and as machine designers establishing new fabrication tools and processes. We have the following design goals:

- **Design freedom.** We want to support developing tool-pathing programs for both established and novel fabrication processes. We want to be able to draw any toolpath and turn it into an executable machine action for a specific machine. Therefore, we need to support many inputs (e.g., geometry) and exports (e.g., G-code flavors).
- **Shared and reusable resources** We want to support ease of implementation and code reuse. Therefore, we need to expose the computational functions that normally are hidden in CAM tools and expose them as reusable blocks that can be adapted, combined, and reused. These modules must support being combined and mixed together.

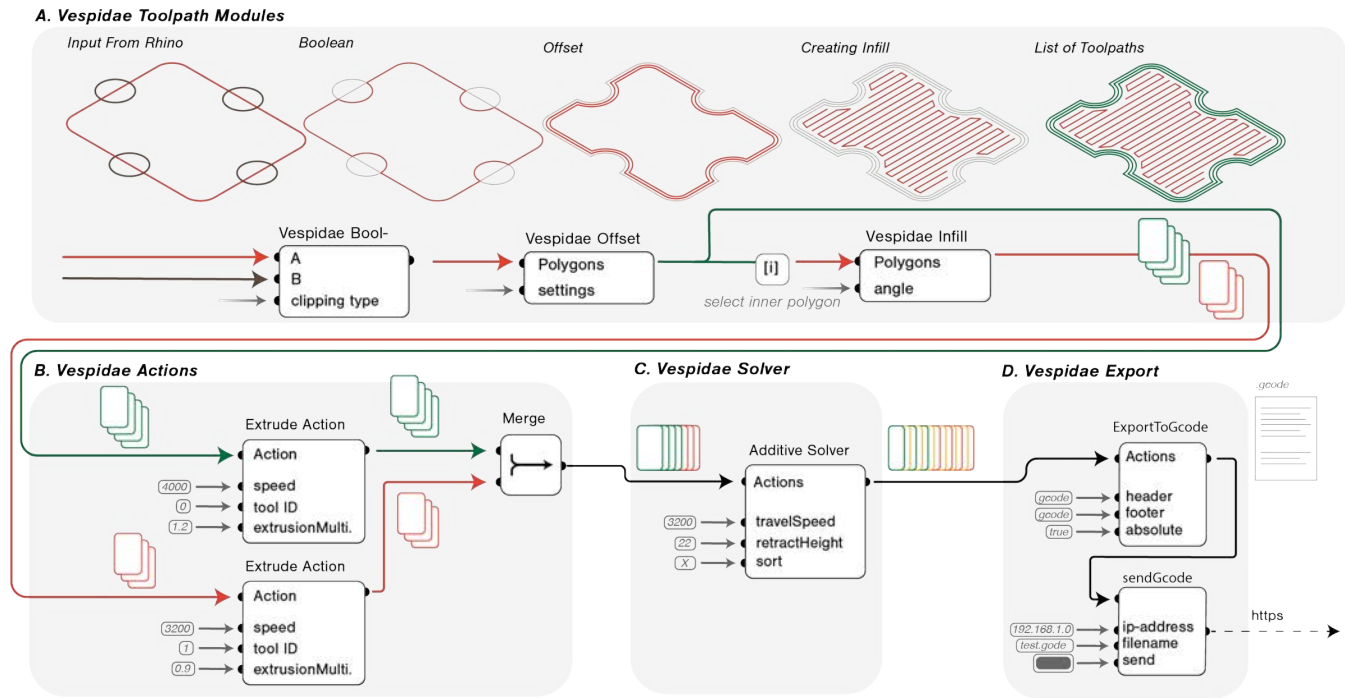


Figure 4: An in-depth look at generating a 3D printing toolpath with a Vespidae program. (A) Vespidae Toolpath: The inputted polygons are separated into two input groups that are used for the boolean operation. The Boolean module performs a clipping operation where B is clipped away from A. The resulting polygon is then offset twice using the Offset module. Infill is generated from the inner of the offset curves using the Infill module. Finally, the offset curves and infill curves are separated into two different data pipelines. **(B) The polygons are tagged with meta-data by Action modules.** In this particular case, the two data pipelines are tagged with Extrude Actions with their own distinct meta-data to distinguish infill parameters from perimeter parameters. **(C) The lists of Actions are sorted in the Solver,** which also generates moves between actions. **(D) Finally, the sorted Actions are converted to G-code and streamed to the machine.**

- **Fast iteration cycle.** The traditional CAD-to-CAM-to-machine workflow requires users to traverse several different software and data formats, which lengthens the iteration cycle and can be cumbersome. Therefore, we choose to implement our program within a CAD environment, eliminating the need to switch software and streamlining the workflow. Furthermore, to shorten the path of exporting and uploading generated machine code, we choose to support direct communication with a machine controller from our program.

These design goals informed our implementation of the Vespidae framework. Specifically, Vespidae provides individual modules that can be mixed together to form the data pipeline required to transform drawn geometry to machine control. By combining these modules, we can form the full stack necessary to go from drawn shapes to machine code and tailor toolpathing programs for a broad palette of different fabrication processes.

Vespidae is implemented as an open-source library in Grasshopper, which is a graphical programming language built into the popular CAD tool Rhino3D. Grasshopper comes with a built-in suite of tools for processing geometry data in Rhino. These tools are organized into different categories, which are called ‘shelves’ in Grasshopper. Vespidae is implemented as its own shelf that users can

import into Grasshopper. Each Vespidae-module is written in C# using the McNeels Template Tool for Visual Studio [45]. We chose Rhino and Grasshopper due to its rich API library of geometry functions and popularity. Figure 3 shows an overview of all the current modules of Vespidae, with additional detail in 1. Vespidae modules are organized into four distinct categories, which each represent a step in the complete computational stack of a toolpathing program:

- **Vespidae Toolpaths** manipulate geometry inputs and generate toolpaths based on the input.
- **Vespidae Actions** tag geometry data with metadata describing fabrication processes (e.g., parameters such as spindle speed when milling or extrusion rate when printing)
- **Vespidae Solvers** convert sequences of Vespidae Actions into specific machine instructions sorted according to user input (e.g., the Additive Solver sorts Actions based on the Z-height of each Action). This category also contains the **Visualize** component, used to visualize Vespidae Actions in Rhino.
- **Vespidae Export** converts Vespidae Actions into machine specific languages (e.g., G-code). It can also communicate with machines or external applications directly over https and websockets.

We emphasize that Vespidae does not enforce a strict way that these modules should be combined. Users are free to mix and match any module from any of the categories to shape their toolpathing program as they see fit, as long as their data input and output match. For example, a user can tag polygons drawn in Rhino with an Action and translate the Action directly to G-code without using any Solver. Alternatively, a user can inject additional G-code commands into a list of pre-existing Vespidae-generated G-code. Furthermore, we have taken extensive steps to document the underlying codebase of Vespidae and include guides for how users can implement their own modules that build on top of Vespidae's functionality.

	Module Name	Description
Toolpath	Offset	Offsets inputted polylines by x number of times with y distance
	Boolean	Boolean operations on polylines.
	Infill	Generate infill within any closed polygon
	Slice	Slices a brep model into layers with x distance
Action	Extrude-Action	Tags inputted curves with Extrude-Action
	Cut-Action	Tags inputted curves with Cut-Actions
	Generic-Action	Tags inputted curves with Generic-Actions
	Probe-Action	Tags inputted points with Probe-Actions
Solvers	Generic Solver	Takes inputted sequence of Actions and outputs complete fabrication program by sorting the Actions and adding Travel-Actions between each Action. Provides several strategies for sorting.
	Additive Solver	Same as above, but specialized for layer-by-layer based processes. Provides optional sorting strategies on each layer.
	Visualize	Visualizes Vespidae Actions. Includes options for what type of Action to visualize and options to visualize toolpath directions using arrows
Export	ExportToGcode	Converts any Action or sequences of Actions into G-code. Provides G-code injection both prior and after generated G-code.
	sendGcode	Sends G-code to specified end-point over HTTPS asynchronous.
	VespidaeStream	Streams Vespidae Actions over specific websocket asynchronous.

Table 1: Vespidae Module Overview. We implemented several modules in each of the four Vespidae categories to support multiple types of digital fabrication, including 3D printing and CNC milling.

4.1 Vespidae Toolpaths

The first step in a toolpathing program is to describe the toolpath itself. In our system, toolpaths are initially represented as polygons. The polygons can be generated in three different ways: (1) by drawing shapes in Rhino and streaming these to Grasshopper, (2) by creating parameterized scripts within Grasshopper to generate polygons, or (3) through a combination of (1) and (2). Simple shapes can be drawn in Rhino and further modified in Grasshopper, which contains a rich ecosystem of modules for manipulating geometry data through built-in and external libraries.

Vespidae includes modules for the most common and important functions found in toolpathing software. The Toolpath category includes modules for polygonal boolean and offsetting operations, generating infill or pocket patterns from within closed polygons, and slicing surface geometry into layers of polygons. These are distinct from other geometry processing modules as they control aspects that are unimportant to regular geometry processing (e.g. rendering), but are important for fabrication. For example, any polyline has a start and an end point. In a render, this distinction is invisible. In digital fabrication, this will determine the direction and order that a machine will execute moves. Therefore, many existing Grasshopper modules are less suitable for manipulating toolpath geometry.

Existing Grasshopper modules perform clipping operations on polygons, but they do not perform well for many common fabrication tasks. Therefore, we implemented our own polygon clipping using the Angus Clipper library [33], leveraging the library's efficiency on large data sets. Furthermore, both Vespidae's Offset and Infill module are optimized for fabrication operations as they are sensitive to inputs that contain nested polygons¹.

Figure 4A shows how the modules from the Vespidae Toolpath category can be used to define toolpaths from a set of input curves imported from Rhino. By combining three modules from Vespidae we are able to set up a procedure that defines the toolpaths for the first layer on a FDM 3D printer. An important note is that the defined program in the example above is not restricted to that particular input geometry. Any additional polygons that are fed into the same data pipeline will be computed similarly with respect to its own shape. Thus users can start with simple shapes to prototype and test their toolpath program, and apply the same logic to additional polygons that are input to the data pipeline.

4.2 Vespidae Actions

In this second category, Vespidae Actions tag polygons with additional meta-data that describes the process we want to assign to them. Prior to being tagged with a Vespidae Action, the polygons only hold the raw geometry data that describes the polygon itself. By tagging the polygons with meta-data, we can encapsulate required information about the operation we want a machine to perform on the polygons. This meta-data can hold information on parameters such as extrusion rate, spindle RPM, and cutting force. Furthermore, Vespidae Actions provide a convenient way to contain multiple operations with different parameters into a single program. When 3D printing, this could mean that there are multiple Extrude-Actions with different speeds or extrusion rates. When using machines that support automated tool changing, completely different processes such as milling and 3D printing can be combined into the same workflow using Actions.

Vespidae currently includes application-specific Action-modules for Cut, Extrude, and Probe. The meta-data includes both common properties like tool-speed or tool-number, and more process-specific properties such as printing temperature and extrusion settings for an Extrude-Action, or spindle speed for a Cut-Action. Additionally, each Action-module includes *code-injection* that enables users to inject raw textual code that will be added to the Actions translation in the subsequent steps. This is explained in more detail in Section 4.4. Additionally, Vespidae includes a Generic-Action module that can be used to generate other types of machine actions without any specialized meta-data.

With Vespidae, it is simple to develop mixed fabrication processes, as we can use different Actions to control different tools starting from the same toolpath. For example, an Extrude-Action that controls an FDM nozzle can be switched with a Generic-Action to control a pneumatically-driven syringe extruder. Through these Action modules, users can freely chain and tag polygons with appropriate meta-data before they are passed along to Vespidae Solvers.

¹Determining inside/outside of polygons within polygons is a crucial step when creating toolpaths from polygonal data. For details see Clipper Documentation [33].

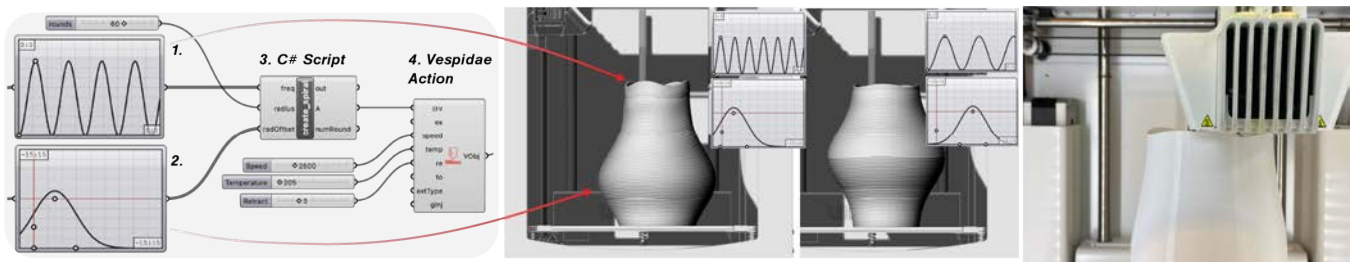


Figure 5: A program for non-planar 3D printing of lampshades using Vespidae. The overall shape of our lamp is defined by two functions we describe in Grasshopper’s *Graph Mapper* components connected to a custom C# script. These functions morph a base spiral shape to change 1: the lamp’s outer profile, and 2: the number of pleats. This lamp geometry, described as one continuous path, is then sent to a Vespidae Action which assigns data that describes how it will be printed. Other Vespidae modules (not shown) handle sorting the Actions and converting the program to machine code. To the right we can see the non-planar toolpath as it’s printing on a Ultimaker 3. Note the top of the print follows an undulating pattern rather than a single z-height.

4.3 Vespidae Solvers

Prior to being processed by a Vespidae Solver, a sequence of Vespidae Actions will only contain the Actions themselves with their associated meta-data. Additional information about how the machine should move between each Action, and in what order, needs to be added to create an executable machine program. The Vespidae Solvers transform chained sequences of Actions into executable machine programs by adding **Travel-Actions** between each Action. A Travel-Action is a move that the machine has to take between each Action. Its path is computed by taking the last point of the previous Action’s path and the first point of the next Action’s path and using these two points to define a path between the Actions. The output of a Solver will be a sequence of Actions with defined Travel-Actions in between them. Thus, the sequence will contain all the information necessary in order for the machine to execute them. If the Action sequence has multiple tools, the machine will execute appropriate tool changes between Actions.

The Vespidae Solvers also incorporate different strategies for sorting the inputted sequence of Actions. As sequences of Actions are chained together in the prior steps, their order is arranged by what order their initial geometry data was generated in or in what order the Actions were inputted to the Solver.

We have implemented two specific solvers that use two different sorting strategies: the **General Solver** and the **Additive Solver**. The General Solver takes any inputted Action (Cut, Extrude, Generic, etc.) and sorts them based on a chosen sort criteria. Actions can be sorted by direction (X, Y, or Z) or by Action type (for example, Extrude before Cut). This enables the user to optimize their execution. If no sorting is chosen, the inputted Actions will be executed in the order that they are inputted in. Here, the Solver will only generate travel moves between the Actions. The Additive Solver is a solver targeted for additive fabrication processes and it filters out any other Action except Extrude-Actions. It then separates the Extrude-Actions into dictionaries with the Z-heights as the indices and sorts all the Actions based on Z-height in ascending order. The Additive Solver also provides sorting options for the order Actions on each layer should be executed (X, Y, Z, or by tool-ID, just like the Generic Solver).

The General Solver can also be used as a template to create custom Solver modules that can enable further fine-tuned control. For example, they could implement additional sorting and optimization methods, insert additional Actions, or perform safety checks given past and future Actions. Furthermore, Vespidae allows multiple Solvers to be chained together for additional functionality.

4.4 Vespidae Export

Finally, Vespidae Export-modules both translate Vespidae Actions into textual machine code, which in most cases is G-code, and communicate with machines outside of Grasshopper.

The **ExportToGcode** module translates sequences of Actions into G-code. Each Vespidae Action holds its own translation routine that describes how the Action should be translated into machine code. The Export component loops through each Action in the inputted list of Actions, checks what type of Action each instance is, executes its given translation routine, and adds the translated machine code to a single output file. If an Action is given additional textual input through the G-code injection input, this will be added to the translation for each Action. The ExportToGcode component also has options for injecting header or footer G-code into the final G-code file, since some machines require a unique header G-code to prime the machine before it can execute its program.

Vespidae contains two modules to enable communication with external end-points. **SendGcode** streams G-code to a defined endpoint over HTTPS. This component was originally developed for the Duet-3 machine controller [15] but is extendable to machine controllers with similar architecture for uploading G-code. A user inputs an IP-address and filename and toggles send to upload a generated G-code file to the controller. The **VespidaeStream** component streams sequences of Vespidae Actions as json objects over a user-defined websocket connection. We implemented this component as a means to work with Vespidae generated data outside of Grasshopper, for example to visualize Action-objects in a browser application. Both the SendGcode and the VespidaeStream components are implemented as asynchronous code using the *GrasshopperAsyncComponent* wrapper originally implemented by Specklesystems [59].



Figure 6: Lampshades printed with Vespidae. The two lampshades to the left were printed with non-planar printing program shown in Figure 5. In that program, we can parametrically vary the pleating, the non-planar layers, and the lamp’s radius at different heights. The rightmost lampshade was printed by extending this program with the *Pulse Extrude-Action* described in Section 5.2, Flexible 3D Prints Through Under-extrusion. This introduced additional transparency and flexibility in the shade. We experimented extensively in order to find good input values for extrusion amount and frequency of pulsing. The challenge was to extrude enough for sticking, yet not so much as to result in a solid print. Being able to quickly export and execute G-code on our printer allowed us to focus on the material exploration aspects rather than on moving files around.

5 DEMONSTRATING VESPIDAE CAPABILITIES

To demonstrate our system we developed material swatches using Vespidae, which we tested as functional objects such as lampshades. Our approach in developing our demonstrations was to iterate on objects in a workflow similar to our usual design practice. This allows us to explore the design space and inform the development of our framework. We selected three demonstrations to highlight specific features of our programming framework. These are non-planar 3D printing, flexible 3D prints through underextrusion, and milled textures. Each highlights unconventional results that can be obtained through low-level control of machine code. These demonstrations form part of the evaluation of our system [40].

5.1 Non-planar 3D printing

FDM 3D printing leaves visible lines on the print where each layer is deposited. Using non-planar printing we can control this surface feature and introduce a material ‘grain’ in different directions. Rather than print one flat planar ‘slice’ at a time, with non-planar 3D printing we can build up shapes through full XYZ control. This is particularly clear in transparent applications such as lampshades. To explore this, we created a toolpathing program that supports generating machine code that supports non-planar 3D printing on an Ultimaker 3. We were particularly interested in gaining fine-grained control of gradient diffusion properties of the lampshade.

To make the toolpath, we created a C# script in Grasshopper that generates a spiralized polygon controlled by a set of input parameters: radius, pitch, and points per circumference, as well as parameters for controlling undulating movement in z axis, cross-sectional shape of lamp and pleat height. The output of the script is a single continuous toolpath. An overview of the script-module and its input parameters is shown in Figure 5.

Using Vespidae, this input toolpath is tagged using an Extrude-Action. We also added an additional Action to the graph for priming the nozzle before executing the print to improve print quality. All of these Actions were assigned to a Solver that generated the travel moves between them. The Solver output goes to a G-Code Convert module. The Ultimaker required a special header file that was injected into Vespidae’s G-Code module.

Using our program, which includes our custom script as well as off-the-shelf Vespidae modules, we could parametrically generate both the geometry and machine instructions for a series of lampshades. Rather than separating this into separate CAD and CAM phases, we’re able to explore the interdependencies between the two. To produce our lampshades, we iterated with parameters including extrusion rate, spiral pitch, and Z-variation that allowed us to explore tactile qualities such as opacity and surface texture. Using visualization tools we were also able to determine what type of geometries were possible without nozzle crashes. This example

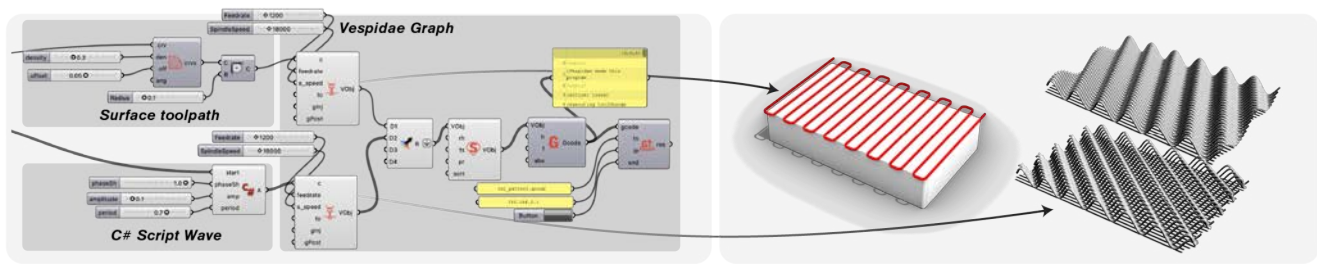


Figure 7: Vespidae modules can be used for several different types of digital fabrication processes, including CNC milling. Shown here is a program for creating undulating surface textures using a CNC mill.



Figure 8: Results from our experiments milling Valchromat using a Vespidae program with Cut-Actions. The ridges, tool marks, and machine speeds and feeds are all parameters we experimented with.

demonstrates the various ways we can tailor and optimize fabrication aspects of our program to a specific geometry, and how we can learn how different geometries impact fabrication outcomes. The lampshades we produced are shown in Figure 6.

5.2 Flexible 3D Prints Through Under-extrusion

Under-extrusion in FDM 3D printing can result in porous prints, a feature that can be taken advantage of to print objects with compliance and flexibility [18]. To explore the material possibilities of under-extrusion, and to demonstrate extending Vespidae with a new module, we developed an Action module, the **Pulse Extrude-Action**. Similar to Forman et al. [18], this module pulses extrusion for a ‘blobby’ effect, introducing a repeating microstructure, which we used to print transparent and flexible lampshades.

There are two steps to implementing new modules in Vespidae. The first step is to generate a new Grasshopper component that will represent the new Action module. In this component, we define the input and output data types and define the function it computes with. Vespidae’s documentation builds on existing component-definition documentation², specifying how to incorporate the Vespidae data formats into the module. The second step is to extend the Vespidae source code with a new Action. Here Vespidae uses C# interfaces that require specific functions to be implemented for the code to compile. This forces developers to define how the Action should translate into G-Code. For our Pulse-Extrude-Action, we add G-Code that extrudes N amount of material at ω frequency, where N and ω are input parameters. With a single new Vespidae

²How to define new components in Grasshopper using C# is documented in detail by McNeel [45].

module, we were able to explore a large design space of flexible 3D prints.

5.3 Milled Textures on Valchromat Wood Composite

Vespidae is not designed exclusively for 3D printing. Its toolpath generation can also be used for other types of digital fabrication such as CNC (computer numerically controlled) milling. The marks that different milling bits leave on a surface can be an interesting artefact to explore and use as a design feature. Where most CNC milling toolpathing software focuses on efficiency for removing material with high precision tolerances, we were interested in exploring how we could use the direction and spacing of milling passes to texture the material, using the toolpaths as an aesthetic feature.

We set out to create a program that would allow us to explore this design space. Like with the non-planar toolpathing, we wrote a custom C# script to generate our toolpath geometry. This approach reflects our own preferences as computational designers—other approaches to generating geometry, such as designing in Rhino or sketching in Illustrator are also valid. Our script constituted a parameterized set of sine waves which let us vary the distance between each pass and their angle. The output of this script was a set of polygons representing our toolpath geometry. We input these into the Vespidae Cut-Action component, which we connected to a Vespidae Solver, which was finally connected to a Vespidae G-code-Export module. Our CNC mill uses standard Marlin-style G-code, which Vespidae supports. This program is shown in Figure 7.

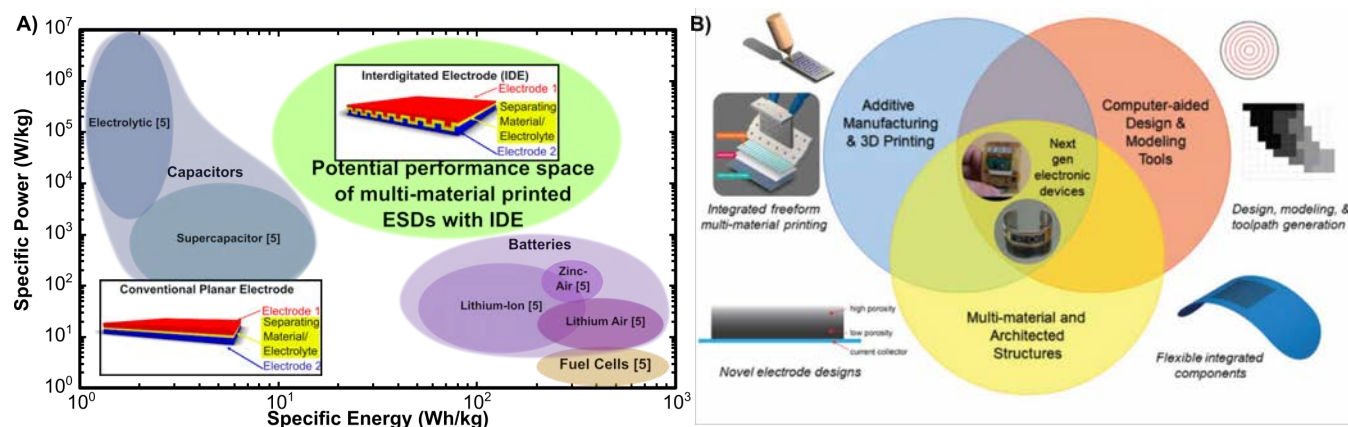


Figure 9: A) A high-level plot showing power and energy densities for various ESDs. Subcategories of these ESDs are shown based on the state-of-the-art in flexible electronics, reported in [5]. These ESDs are typically made using planar electrodes (inset, bottom left). By printing ESDs with IDE structures (inset, top right), ESDs can reach new areas of energy and power density, highlighted here in green. B) Future vision for 3D printing powered portable, flexible electronics at the intersection of CAD/CAM, additive manufacturing, and multi-material architected structures. Reprinted from [8]. © The Electrochemical Society. Reproduced by permission of IOP Publishing. All rights reserved. This highlights the need for further exploration of human-machine collaboration.

We cut all the pieces in Valchromat [30], a medium-density fibreboard (MDF) that comes in deeply dyed colors. We calculated an initial feedrate and cut depth using data on our milling bit and material. We experimented with speeds and depths through several test cuts in order to understand the effects of different settings. To do these tests, we input different speed and depth parameters into Vespidae Cut-Actions. After building our intuition on milling speeds and depths with these tests, we focused on the possibilities of pattern. By varying the sine waves of our input geometry, we tested many variations, exploring how the pass spacing and ridge height played against each other. We conducted our tests with one end-mill geometry. Exploring how different end-mills can be used to texture material would be our next step. Outcomes of our material testing are shown in Figures 8 and 1.

This workflow also highlights a potential danger. Whereas a 3D printer is more forgiving when it comes to problematic G-Code, a milling machine works with strong forces and high speeds. Toolpaths require checking and validation to ensure that the machine is not going to be conduct a dangerous move. Vespidae is not unique in that it allows users to write dangerous toolpaths—most CAM software permits this. However, Vespidae purposefully reduces the friction to running different toolpaths on machines. This makes it easier to quickly run a program without a thorough check. We join calls for future research into managing safe human-machine collaboration [41], such as using additional “Toolpath Style Sheets” [64] to visualize dangerous actions.

These three examples demonstrate some of the possibilities that Vespidae introduces with readily available digital fabrication equipment. Beyond off-the-shelf 3D printers and milling machines, we believe that Vespidae can also support the development of more experimental processes using custom machines and novel materials. In the next section, we describe a more in-depth exploration using Vespidae for experimental workflow development.

6 USING VESPIDAE FOR 3D PRINTING ENERGY STORAGE DEVICES

We conducted an in-depth study where one of the authors, a materials engineer (ME), used Vespidae for materials research. We structured our study as a collaboration between the authors drawing from a Research through Design (RtD) approach. The ME has a background in mechanical design and materials, so they were comfortable using CAD and traditional 3D printing slicers but had limited experience in software development. The ME’s prior software experience involved using Python for data visualization and writing simple Arduino programs. The ME’s goal was to develop custom toolpaths for 3D printing energy storage devices (ESDs), which broadly refer to rechargeable and non-rechargeable batteries, capacitors, and fuel cells (Figure 9). Energy researchers such as the ME are currently exploring how the combination of new materials and fabrication processes can produce ESDs with complex geometric structures to increase their performance [5, 8]. In this research, the development of new materials and appropriate fabrication processes are highly interdependent. Exploring these interdependencies demands full design control over the fabrication process. The ME needed software that provided control over every toolpath for their custom-built machine while facilitating quick iteration cycles to enable efficient exploration of different material and structural configurations for ESDs. These requirements and constraints informed our development and documentation of Vespidae.

6.1 Methods

The study took place over a time span of six months, with the ME using Vespidae a few hours per week on average. We discussed progress in bi-weekly virtual meetings, and tracked the ME’s stored updates on Github which included their programs and notes about

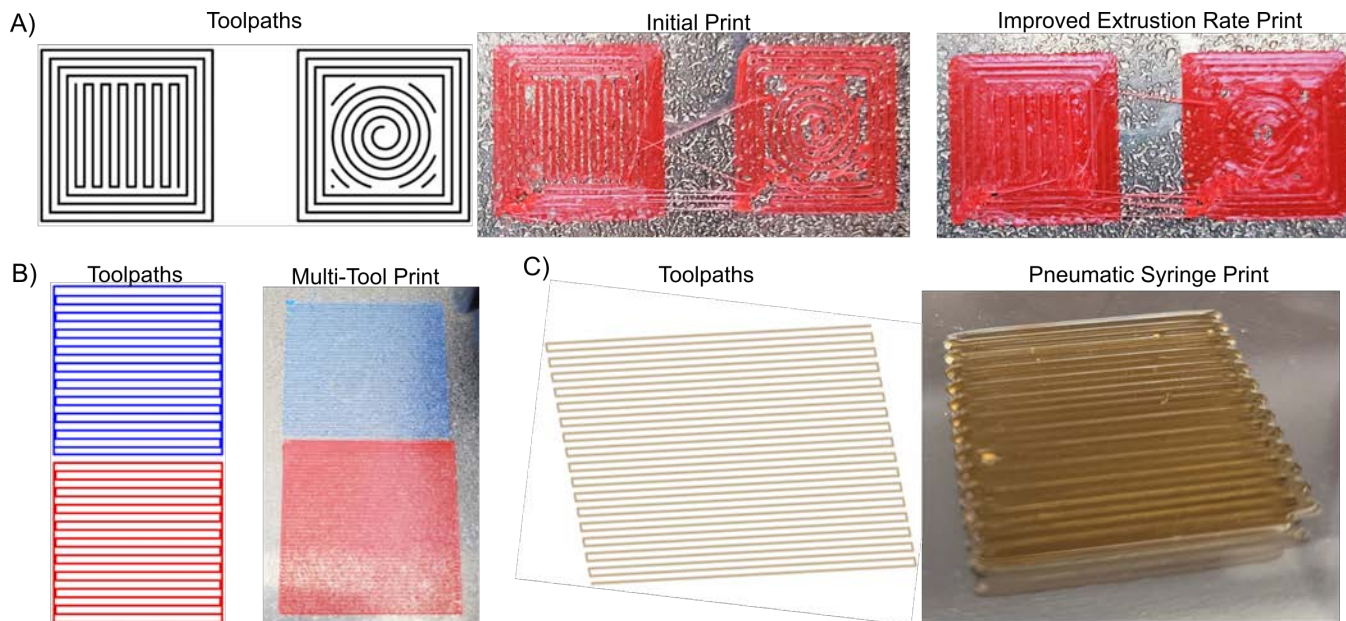


Figure 10: Experimental Vespidae toolpaths. A) Single-layer, single-tool toolpaths to optimize extrusion multipliers. Toolpaths are tagged with Extrude-Actions and sent through a Solver, before being uploaded directly to the printer. Test prints varied extrusion multipliers from 0.9-3 and printing speeds from 500-2000 mm/min. B) Single-layer, multi-tool toolpaths to test toolchanging. Toolpaths were tagged with an Extrude-Action for each tool and merged into a single Solver. C) Single-layer, single-tool toolpaths to test pneumatic syringe control. Sample printed at 700 mm/min at 20 psi.

their interactions with Vespidae. Over the course of the study, we added updates and new features in the Github repository. We used insights from the collaboration to develop additional documentation and refine aspects of the Vespidae framework.

The ME conducted their experiments in an academic research lab, focusing on clean energy using a custom-built multimaterial 3D printer. The printer was based on an open-source hardware design with a toolchanger that can switch between different end-effectors [69]. The printer used both mechanically-driven heated extruders and pneumatically-driven syringe extruders. A Duet3 machine controller ran the printer and was connected to a computer over TCP. With permission, we have reproduced quotes from our meetings and details of their experiments.

6.2 Context and Goals for 3D Printing Energy Storage Devices

At a basic level, ESDs all operate on electrochemical reactions that involve shuffling and storing ions between two different electrodes and managing the flow of electrons through an external circuit [61]. As shown in the bottom-left inset of Figure 9A, a typical structure for most ESDs consists of two flat electrodes, current collectors attached to each electrode, and a separating sheet inserted between the two electrodes to prevent physical contact that can lead to device failure. ESDs must rapidly transport ions and electrons to yield high-energy (Wh/kg) and high-power (W/kg) density performance. The growing demand for long-duration (increased Wh/kg) and high-performing (increased W/kg) portable, flexible electronic applications [5] has motivated a shift towards developing compact

ESDs with more complex 3D electrode structures. These complex 3D structures cannot easily be created using existing CAD, CAM and additive manufacturing tools, and researchers in the field have called for research to span these gaps (Figure 9B) [8].

Researchers in the battery field note that the typical flat electrodes experience a tradeoff between high power (W/kg) and high energy (Wh/kg) because electrode thickness is the main parameter that can be changed in this structure [9, 28, 43]. With this tradeoff, thicker electrodes have slower ion transport that leads to lower power density, while thinner electrodes lead to lower energy density. One solution to increase power and energy density at the same time is to create ESDs with an interdigitated electrode (IDE) structure [9, 17, 28, 29, 43]. IDEs have two electrodes with intricate 3D column structures arranged in an interlocking pattern (Figure 9A, inset top right). However, most ESDs with IDE structure have to date been fabricated with semiconductor processes [29]. Semiconductor processing is expensive and time-consuming. The fabrication process cannot be easily altered for different IDE dimensions or alternate ESD structures, limiting its feasibility in a field where materials and electrode designs are constantly being altered for new applications.

An alternative approach to fabricating ESDs with complex 3D structures is the combination of multi-material 3D printing and adaptable toolpathing software. 3D printing provides the ability to print a wide range of materials in freeform geometries to enable complex multi-material structures such as IDEs [44]. Meanwhile, adaptable toolpathing provides the flexibility to tweak or redesign a structure by altering toolpaths in CAD. Adaptable toolpathing

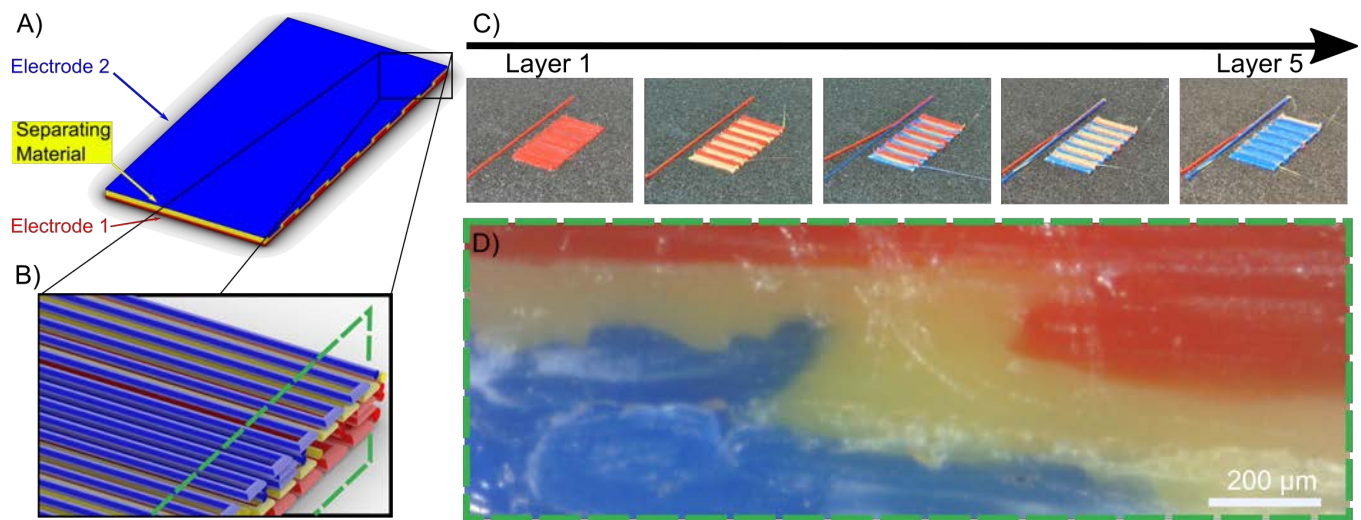


Figure 11: Printing process of an interdigitated ESD. A) Conceptual CAD Model of interdigitated ESD, B) Closeup representation of toolpath visualization for ESD, C) Optical images of printed ESD after each layer of printing, which is made using three mechanically-driven heated extruders loaded with different colors of PLA printed with extrusion multipliers varying from 1.0-1.7, D) optical image of a cross-section of the resulting interdigitated ESD, cut along the path shown in B.

software is needed to support the efficient exploration of new materials, hardware, and ESD structure design for printing new ESDs with complex 3D structures. Previous work on multi-material 3D printing has required either time-intensive manual code generation [66] or software specialized for a specific application that cannot be easily adapted to other processes [6, 55]. Vespidae addresses limitations of prior work, making it a promising software framework for supporting exploration into the challenges of multi-material printing of ESDs.

6.3 Experiments in 3D Printing IDE Structures using Vespidae

Printing IDE structures requires printing three different materials, where each material acts as either an electrode or a separating material. The materials were printed with different extruders on the machine. During the study, the ME printed with test materials that mimic the mechanical and printing properties of electrode and separator materials. This allowed the ME to focus on toolpath generation rather than materials handling. They used polylactic acid (PLA) filaments for the mechanically-driven heated extruder and a gel for the pneumatically-driven syringe extruder.

To familiarize themselves with Vespidae, the ME first printed PLA using a single extruder. Their process is shown in Figure 10A. At this point, it was interesting to observe how the machine became an active element in the development of their toolpath programs. For example, they did several test prints to improve their print quality by modifying their extrusion multipliers and print speed parameters. These tests varied the extrusion multipliers from 0.9-3 and the print speed from 500-3000 mm/min. They remarked that the workflow provided them with a “tight feedback loop” where they could make small edits and run the code on their machine without

having to toggle between multiple programs like they previously did with traditional slicers.

Vespidae really helps with my process and gives me a lot more granular control of my designs, which wouldn't be possible with standard slicers. I'm now able to iterate quickly and efficiently, without the need to juggle multiple programs just to try small tweaks.

Once they developed a satisfactory program using one extruder, they extended the program to multiple extruders. As seen in Figure 10B, the ME conducted several tests to experiment with different extrusion parameters for extrusion and execution order of the different extruders in their setup. This was an important part of their testing because the order of material deposition impacts IDE structure and ESD functionality. The ME mentioned that controlling this execution order would be infeasible with traditional slicers.

They also printed with pneumatically-driven syringe extruder, which is controlled with different logic than a mechanically-driven extruder. The ME was interested in eventually being able to print with both extruders in one workflow, which they could not do with their off-the-shelf slicer. They performed initial tests, seen in Figure 10C, by using a Generic-Action with injected G-code to control the extruder instead of the Extrude-Action, printing at 700 mm/min.

The ME then moved on to work with more complex multi-layered prints that were closer to a printed IDE structure using three colors of PLA (Figure 11A). The ME first drew multiple layers of toolpath geometry in Rhino (Figure 11B), and piped this data through the same pipelines used in their previous multi-tool prints. Layer-by-layer images and a cross-section of the final print (Figure 11C-D) shows that the ME successfully printed a multi-material IDE structure using Vespidae. The ME noted that to make this IDE structure, they used Vespidae to adjust the extrusion multipliers for each toolpath and specify the execution order for each layer.

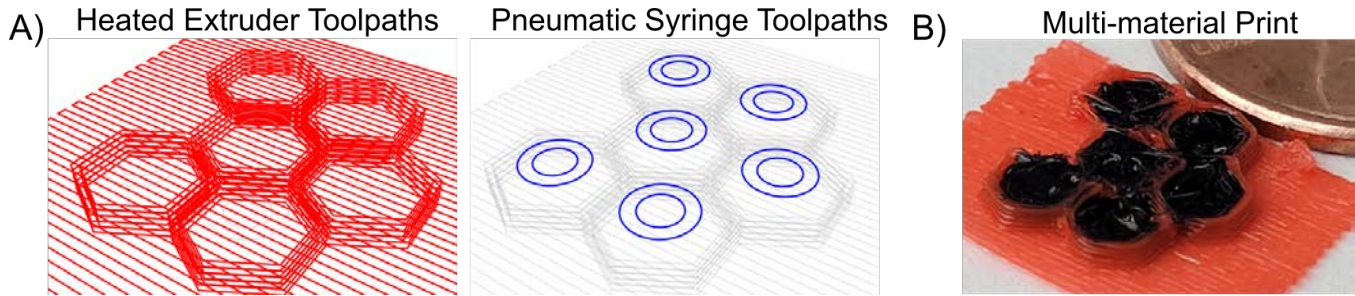


Figure 12: A) Toolpaths for a multi-material print. Multi-layer hexagonal structures were first printed with PLA from a mechanically-driven heated extruder. The hexagons were then filled in using a pneumatically-driven syringe extruder. B) The resulting multi-material print. Edge of penny shown for reference.

This demonstrated how Vespidae’s design freedom helped the ME improve the quality of their IDE structures.

They remarked about how designing in raw toolpaths made them reflect differently about the overall design of the IDE structure:

Having the ability to tune the finer details in my tool-path also makes me more closely consider my designs and examine how I can improve my devices through toolpath optimization.

The ME then began testing different multi-material printing techniques. Their goal was to explore the range of toolpath generation possibilities with Vespidae and build up expertise with their materials to help them design better future toolpaths. For one experiment, the ME replaced the PLA separating material with gel printed from a pneumatically-driven syringe extruder. These prints were unsuccessful because of poor adhesion between the PLA and gel, but the ME was able to easily switch between extruder types by swapping one Extrude-action for a Generic-Action with injected code. For another experiment, they printed a honeycomb structure filled with gel (Figure 12). These tests demonstrate how Vespidae encourages experimentation and enables new workflows that combine traditional layer-by-layer printing with other techniques.

A limitation was the need for clearly visualizing toolpaths. Vespidae has tools for filtering Actions based on properties in its metadata. However, as the number of Actions increased it was harder to see how the Actions were structured for execution. The ME used the list tools in Grasshopper, where they iterated through each list of Actions to verify that their execution order was as they expected. They remarked that better visualization would be an improvement.

By the end of the study we also observed that the ME started to extend the functionality of Vespidae by implementing their own Vespidae modules in Grasshopper. Some of the Toolpath modules were developed in dialogue with the ME. For example, Vespidae Toolpaths initially only included modules to use the Clipper library to perform offset and boolean operations. However, as they were using Vespidae to make more complex toolpaths, it became clear that they needed more infill functionality. During the study, multiple functions were added to Vespidae, including infill, new methods of sorting toolpaths, and pneumatic extruder control.

After the six-month study, the ME was successfully able to develop custom toolpaths for printing IDE structures using representative test materials. The ME remarked that Vespidae made this task

easier compared to conventional slicers, because they were able to fine-tune individual toolpaths and more quickly iterate between small tweaks to the parameters. The ME also emphasized that they would not have been confident in either hacking a slicer or writing their own program from scratch for developing the toolpaths necessary for the multi-material and gel-filling prints in their later experiments. By the end of exploration, the ME became confident with using Vespidae for developing unconventional toolpaths for their research, and was comfortable with developing new features, and begun developing more Vespidae modules for a different printer that used different tools and ran on different machine code.

7 DISCUSSION

Jacobs et al. [32] called upon HCI researchers to consider not just efficiency and ease-of-use in technology, but also how digital technology can function as an expressive medium. For digital fabrication, they highlight “building expressive computational tools”, “blending the digital and the physical”, and “research through artifact creation or in-the-wild studies”. This call for expressivity in digital fabrication is echoed broadly [3, 23, 35, 37, 65]. With Vespidae, we aimed to heed these calls by contributing both a system that invites iteration and our own experiments with it. With this, we seek to participate in an ongoing research effort to better understand the opportunities of computer-controlled machines, especially given broadening access to these tools through makerspaces, affordable hobbyist equipment, [4, 42] and “maker” attitudes [56]. Here, we explicitly reject slick user experiences in favor of systems that embody “the pliable machine” [37]: alterable machines, inviting unintended use, and a culture of experimentation and sharing.

Systems research evaluation has historically considered usability and novelty key assessment criteria to mixed ends [14, 24, 40]. We’d likely not fare well along either metric. In our case, we instead evaluate our contributions by considering their “pliability”. In this paper, we demonstrated how Vespidae modules might be mixed, matched, and extended. Through our lampshades and milled samples, we sought to clearly show how our system is designed for pliability. Each of our demonstrations draws from our own computational craft practice; here we are conducting open-ended material explorations that are primarily aesthetically-driven. Rather than considering outcomes of this exploration as mere variations using

parameters the Vespidae system provides, we'd like to consider them as their own "loose ends" or "swatches" [22, 50].

Furthermore, we hypothesized that a concrete, real-world fabrication challenge would push our system to truly support the messiness of process development. By collaborating on interdigitated energy storage device toolpaths on an experimental machine, we were able to push the boundaries of current practice guided by real-world constraints. Our emphasis on maximizing power and energy density may seem like a departure from other explorations that emphasize craftsmanship. However, here we echo a rich prior work in HCI that draws parallels between craft and engineering [7, 11, 16]. We argue that our collaboration demonstrates how other fields, such as materials engineering, can draw from HCI methods.

Finally, one might argue that our system and experiments are limited to expert practice. After all, beyond Vespidae, our examples include custom code and machines, neither of which could easily be replicated by novices. However, we think this criticism relies on an overly simple construction of expertise. Digital fabrication encompasses code, materials, craft, and hardware and is used for a wide range of applications—what Somanath et al. [57] call a technological assemblage. Making this more novice-friendly greatly depends on what the humans in that assemblage are novices in. Therefore, we argue that systems such as Vespidae that are designed for reconfiguration and change can support a broader range of practitioners than those which streamline workflows with strong abstractions. As Resnick et al. [53] argued, *"the higher-level the primitives, the easier they are to use, but the less they can do"*.

In Section 6 we showed how rapid iteration with simple tests allowed the materials engineer to quickly build an understanding of their custom system. Their established understanding let them design their own abstractions (in the form of Vespidae modules) which didn't directly correspond to existing modules but were appropriate for their system. We argue that this is an example of supporting the learning of anticipation with fabrication systems. Here, exploration and iteration gives insight into the details of fabrication, building trust in how it works, and then understanding the power of the abstractions the system provides. Ultimately, we believe this type of exploration with material, machine, and code will push the boundaries of what is possible in digital fabrication, generating impact that reaches beyond HCI.

8 CONCLUSION

We presented Vespidae, a framework which supports creating programs for toolpath generation for digital fabrication machines. Vespidae provides a library of modules for authoring toolpathing software for a broad range of fabrication processes including 3D printing and CNC milling, interfaces for direct machine control, and templates for custom modules. Built-in modules include Toolpath, Action, Solver, and Export types for manipulating geometry, adding process-specific metadata, queuing and sorting actions, and running these on specific fabrication machines. We demonstrate how Vespidae supports exploratory and unconventional fabrication through example programs for non-planar 3D printing, underextrusion for flexible prints, and CNC milling textures. We furthermore describe an in-depth study with a materials engineer who is developing multimaterial 3D printing processes for architected structures

for energy storage devices. We show that Vespidae enables rapid iteration and experimentation across geometry and manufacturing process.

We argue that Vespidae encourages the exploration of the boundaries of a design space established by material and machine. By supporting the authoring of workflows that enable inspection and modification at each step, we scaffold the development of new fabrication processes and practices for applications ranging from the performant to the aesthetic. Here, rather than user-friendly and streamlined fabrication workflows that can build very little, we support a broad range of creators with small, flexible modules from which we hope to build much more.

ACKNOWLEDGMENTS

We would like to acknowledge Jens Dyvik for his insights into fabrication workflows and the inner workings of toolpathing software. We would like to thank Jennifer Jacobs and Shenna Shim for feedback on drafts of this manuscript. This work has been partially funded by NSF Award #2007045 and a Defense Advanced Research Projects Agency (DARPA) Young Faculty Award and Director's Fellowship under grant number D19AP00038. The views, opinions, and findings expressed in this work are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government, and no official endorsement should be inferred. This is approved for public release and distribution is unlimited.

REFERENCES

- [1] Lea Albaugh, Scott Hudson, and Lining Yao. 2019. Digital Fabrication of Soft Actuated Objects by Machine Knitting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3290605.3300414>
- [2] Celena Alcock, Nathaniel Hudson, and Parmit K. Chilana. 2016. Barriers to Using, Customizing, and Printing 3D Designs on Thingiverse. In *Proceedings of the 19th International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) (GROUP '16). Association for Computing Machinery, New York, NY, USA, 195–199. <https://doi.org/10.1145/2957276.2957301>
- [3] Kristina Andersen, Ron Wakkary, Laura Devendorf, and Alex McLean. 2019. Digital Crafts-Machine-Ship: Creative Collaborations with Machines. *Interactions* 27, 1 (dec 2019), 30–35. <https://doi.org/10.1145/3373644>
- [4] Gabrielle Benabdallah, Sam Bourgault, Nadya Peek, and Jennifer Jacobs. 2021. Remote Learners, Home Makers: How Digital Fabrication Was Taught Online During a Pandemic. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 350, 14 pages. <https://doi.org/10.1145/3411764.3445450>
- [5] Yvan Bonnassieux, Christoph J Brabec, Yong Cao, Tricia Breen Carmichael, Michael L Chabiny, Kwang-Ting Cheng, Gyoujin Cho, Anjung Chung, Corie L Cobb, Andreas Distler, et al. 2021. The 2021 flexible and printed electronics roadmap. *Flexible and printed electronics* 6, 2 (2021), 023001. <https://doi.org/10.1088/2058-8585/abf986>
- [6] Gilbert T Carranza, Ubaldo Robles, Cesar L Valle, Jesus J Gutierrez, and Raymond C Rumpf. 2019. Design and Hybrid Additive Manufacturing of 3-D/Volumetric Electrical Circuits. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 9, 6 (2019), 1176–1183. <https://doi.org/10.1109/TCPM.2019.2892389>
- [7] Amy Cheadle and Steven J. Jackson. 2015. Digital Entanglements: Craft, Computation and Collaboration in Fine Art Furniture Production. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Vancouver, BC, Canada) (CSCW '15). Association for Computing Machinery, New York, NY, USA, 958–968. <https://doi.org/10.1145/2675133.2675291>
- [8] Corie L Cobb and Christine C Ho. 2016. Additive manufacturing: rethinking battery design. *The Electrochemical Society Interface* 25, 1 (2016), 75. <https://doi.org/10.1149/2.F08161if>
- [9] Corie L Cobb and Scott E Solberg. 2017. Communication—Analysis of Thick Co-Extruded Cathodes for Higher-Energy-and-Power Lithium-Ion Batteries. *J.*

- Electrochem. Soc.* 164, 7 (2017), A1339. <https://doi.org/10.1149/2.0101707jes>
- [10] Audrey Desjardins and Timea Tihanyi. 2019. ListeningCups: A Case of Data Tacitility and Data Stories. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) (*DIS '19*). Association for Computing Machinery, New York, NY, USA, 147–160. <https://doi.org/10.1145/3322276.3323694>
 - [11] Laura Devendorf, Katya Arquilla, Sandra Wirtanen, Allison Anderson, and Steven Frost. 2020. Craftspeople as Technical Collaborators: Lessons Learned through an Experimental Weaving Residency. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376820>
 - [12] Laura Devendorf and Kimiko Ryokai. 2015. Being the Machine: Reconfiguring Agency and Control in Hybrid Fabrication. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (*CHI '15*). Association for Computing Machinery, New York, NY, USA, 2477–2486. <https://doi.org/10.1145/2702123.2702547>
 - [13] Mustafa Doga Dogan, Faraz Faruqi, Andrew Day Churchill, Kenneth Friedman, Leon Cheng, Sriram Subramanian, and Stefanie Mueller. 2020. G-ID: Identifying 3D Prints Using Slicing Parameters. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376202>
 - [14] Paul Dourish. 2019. User Experience as Legitimacy Trap. *Interactions* 26, 6 (oct 2019), 46–49. <https://doi.org/10.1145/3358908>
 - [15] Duet3D. 2022. Duet3D designs hardware, firmware and software for 3d printers, CNC machines and a lot more. Retrieved Jan 12, 2023 from <https://www.duet3d.com>
 - [16] Mike Eisenberg and Ann Eisenberg. 1998. Shop Class for the Next Millenium: Education through Computer-Enriched Handicrafts. *Journal of Interactive Media in Education* 1998, 8 (1998), 1–30.
 - [17] Etienne Eustache, Camille Douard, Arnaud Demortiere, Vincent De Andrade, Mylene Vrachet, Jean Le Bideau, Thiery Brousse, and Christophe Lethien. 2017. High Areal Energy 3D-Interdigitated Micro-Supercapacitors in Aqueous and Ionic Liquid Electrolytes. *Advanced Materials Technologies* 2, 10 (2017), 1700126. <https://doi.org/10.1002/admt.201700126>
 - [18] 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* (Virtual Event, USA) (*UIST '20*). Association for Computing Machinery, New York, NY, USA, 1222–1233. <https://doi.org/10.1145/3379337.3415876>
 - [19] Frikk Fossdal, Rogardt Heldal, and Nadya Peek. 2021. Interactive Digital Fabrication Machine Control Directly Within a CAD Environment. In *Proceedings of the 6th Annual ACM Symposium on Computational Fabrication* (Virtual Event, USA) (*SCF '21*). Association for Computing Machinery, New York, NY, USA, Article 8, 15 pages. <https://doi.org/10.1145/3485114.3485120>
 - [20] Frikk H. Fossdal, Jens Dyvik, Jakob Anders Nilsson, Jon Nordby, Torbjørn Nordvik Helgesen, Rogardt Heldal, and Nadya Peek. 2020. Fabricatable Machines: A Toolkit for Building Digital Fabrication Machines. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Sydney NSW, Australia) (*TEI '20*). Association for Computing Machinery, New York, NY, USA, 411–422. <https://doi.org/10.1145/3374920.3374929>
 - [21] Andrew Gleadall. 2021. FullControl GCode Designer: Open-source software for unconstrained design in additive manufacturing. *Additive Manufacturing* 46 (2021), 102109. <https://doi.org/10.1016/j.addma.2021.102109>
 - [22] Bruna Goveia da Rocha, Kristina Andersen, and Oscar Tomico. 2022. Portfolio of Loose Ends. In *Designing Interactive Systems Conference* (Virtual Event, Australia) (*DIS '22*). Association for Computing Machinery, New York, NY, USA, 527–540. <https://doi.org/10.1145/3532106.3533516>
 - [23] Bruna Goveia da Rocha, Johannes M. L. van der Kolk, and Kristina Andersen. 2021. Exquisite Fabrication: Exploring Turn-Taking between Designers and Digital Fabrication Machines. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 434, 9 pages. <https://doi.org/10.1145/3411764.3445236>
 - [24] Saul Greenberg and Bill Buxton. 2008. Usability Evaluation Considered Harmful (Some of the Time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (*CHI '08*). Association for Computing Machinery, New York, NY, USA, 111–120. <https://doi.org/10.1145/1357054.1357074>
 - [25] Liang He, Huaishu Peng, Michelle Lin, Ravikanth Konjeti, François Guimbretière, and Jon E. Froehlich. 2019. Ondulé: Designing and Controlling 3D Printable Springs. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (*UIST '19*). Association for Computing Machinery, New York, NY, USA, 739–750. <https://doi.org/10.1145/3332165.3347951>
 - [26] Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (*CHI '16*). Association for Computing Machinery, New York, NY, USA, 384–396. <https://doi.org/10.1145/2858036.2858266>
 - [27] 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* (Toronto, Ontario, Canada) (*CHI '14*). Association for Computing Machinery, New York, NY, USA, 459–468. <https://doi.org/10.1145/2556288.2557338>
 - [28] Chih-Hsuan Hung, Srikanth Allu, and Corie L Cobb. 2021. Modeling Current Density Non-Uniformities to Understand High-Rate Limitations in 3D Interdigitated Lithium-ion Batteries. *J. Electrochem. Soc.* 168, 10 (2021), 100512. <https://doi.org/10.1149/1945-7111/ac2ac5>
 - [29] Janet I Hur, Leland C Smith, and Bruce Dunn. 2018. High Areal Energy Density 3D Lithium-Ion Microbatteries. *Joule* 2, 6 (2018), 1187–1201. <https://doi.org/10.1016/j.joule.2018.04.002>
 - [30] Investwood. 2022. Valchromat, A Forest of Colour. Retrieved Dec 12, 2022 from <https://www.investwood.pt/en/valchromat/>
 - [31] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial Mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (*UIST '16*). Association for Computing Machinery, New York, NY, USA, 529–539. <https://doi.org/10.1145/2984511.2984540>
 - [32] Jennifer Jacobs, David Mellis, Amit Zoran, Cesar Torres, Joel Brandt, and Theresa Jean Tanenbaum. 2016. Digital Craftsmanship: HCI Takes on Technology as an Expressive Medium. In *Proceedings of the 2016 ACM Conference Companion Publication on Designing Interactive Systems* (Brisbane, QLD, Australia) (*DIS '16 Companion*). Association for Computing Machinery, New York, NY, USA, 57–60. <https://doi.org/10.1145/2908805.2913018>
 - [33] Angus Johnson. 2022. Angus Clipper Library. Retrieved Aug 10, 2022 from <http://www.angusj.com/clipper2/Docs/Overview.htm>
 - [34] Hsin-Liu Cindy Kao, Abdelkareem Bedri, and Kent Lyons. 2018. SkinWire: Fabricating a Self-Contained On-Skin PCB for the Hand. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 116 (sep 2018), 23 pages. <https://doi.org/10.1145/3264926>
 - [35] 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 Annual ACM Symposium on Computational Fabrication* (Cambridge, Massachusetts) (*SCF '18*). Association for Computing Machinery, New York, NY, USA, Article 5, 10 pages. <https://doi.org/10.1145/3213512.3213518>
 - [36] Sophie Landwehr Sydow, Martin Jonsson, and Jakob Tholander. 2020. Machine Sensibility: Unpacking the Embodied and Situated Dimensions of 3D Printing. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society* (Tallinn, Estonia) (*NordiCHI '20*). Association for Computing Machinery, New York, NY, USA, Article 53, 13 pages. <https://doi.org/10.1145/3419249.3420166>
 - [37] Sophie Landwehr Sydow, Martin Jonsson, and Jakob Tholander. 2022. Modding the Pliable Machine: Unpacking the Creative and Social Practice of Upkeep at the Makerspace. In *Creativity and Cognition* (Venice, Italy) (*C&C '22*). Association for Computing Machinery, New York, NY, USA, 220–233. <https://doi.org/10.1145/3527927.3532804>
 - [38] Gierad Laput, Xiang 'Anthony' Chen, and Chris Harrison. 2015. 3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) (*UIST '15*). Association for Computing Machinery, New York, NY, USA, 593–597. <https://doi.org/10.1145/2807442.2807484>
 - [39] Eldy S. Lazaro Vasquez, Netta Ofer, Shanel Wu, Mary Etta West, Mirela Alistar, and Laura Devendorf. 2022. Exploring Biofoam as a Material for Tangible Interaction. In *Designing Interactive Systems Conference* (Virtual Event, Australia) (*DIS '22*). Association for Computing Machinery, New York, NY, USA, 1525–1539. <https://doi.org/10.1145/3532106.3533494>
 - [40] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–17. <https://doi.org/10.1145/3173574.3173610>
 - [41] 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* (Cambridge, Massachusetts) (*SCF '17*). Association for Computing Machinery, New York, NY, USA, Article 11, 2 pages. <https://doi.org/10.1145/3083157.3096344>
 - [42] Silvia Lindtner, Garnet D. Hertz, and Paul Dourish. 2014. Emerging Sites of HCI Innovation: Hackerspaces, Hardware Startups & Incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (*CHI '14*). Association for Computing Machinery, New York, NY, USA, 439–448. <https://doi.org/10.1145/2556288.2557132>
 - [43] Jeffrey W Long, Bruce Dunn, Debra R Rollison, and Henry S White. 2020. 3D Architectures for Batteries and Electrodes. *Advanced Energy Materials* 10, 46

- (2020), 2002457. <https://doi.org/10.1002/aenm.200202457>
- [44] Zhiyang Lyu, Gwendolyn JH Lim, Justin Koh, Yi Li, Yanwen Ma, Jun Ding, Jinlan Wang, Zheng Hu, John Wang, Wei Chen, et al. 2021. Design and manufacture of 3D-printed batteries. *Joule* 5, 1 (2021), 89–114. <https://doi.org/10.1016/j.joule.2020.11.010>
- [45] McNeel. 2022. Grasshopper Guides: Installing tools. Retrieved Sept 30, 2022 from <https://developer.rhino3d.com/guides/grasshopper/installing-tools-windows/>
- [46] David Mellis, Sean Follmer, Björn Hartmann, Leah Buechley, and Mark D. Gross. 2013. FAB at CHI: Digital Fabrication Tools, Design, and Community. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (Paris, France) (CHI EA '13). Association for Computing Machinery, New York, NY, USA, 3307–3310. <https://doi.org/10.1145/2468356.2479673>
- [47] Jifei Ou, Gershon Dublon, Chin-Yi Cheng, Felix Heibeck, Karl Willis, and Hiroshi Ishii. 2016. Cillia: 3D Printed Micro-Pillar Structures for Surface Texture, Actuation and Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 5753–5764. <https://doi.org/10.1145/2858036.2858257>
- [48] Nadya Peek, James Coleman, Ilan Moyer, and Neil Gershenfeld. 2017. Cardboard Machine Kit: Modules for the Rapid Prototyping of Rapid Prototyping Machines. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 3657–3668. <https://doi.org/10.1145/3025453.3025491>
- [49] Hannah Perner-Wilson, Leah Buechley, and Mika Satomi. 2010. Handcrafting Textile Interfaces from a Kit-of-No-Parts. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction* (Funchal, Portugal) (TEI '11). Association for Computing Machinery, New York, NY, USA, 61–68. <https://doi.org/10.1145/1935701.1935715>
- [50] Hannah Perner-Wilson and Irene Posch. 2022. How Tangible is TEI? Exploring Swatches as a New Academic Publication Format. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Daejeon, Republic of Korea) (TEI '22). Association for Computing Machinery, New York, NY, USA, Article 55, 4 pages. <https://doi.org/10.1145/3490149.3503668>
- [51] Isabel P. S. Qamar, Rainer Groh, David Holman, and Anne Roudaut. 2018. HCI Meets Material Science: A Literature Review of Morphing Materials for the Design of Shape-Changing Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–23. <https://doi.org/10.1145/3173574.3173948>
- [52] Christian Remy, Lindsay MacDonald Vermeulen, Jonas Frich, Michael Mose Biskjaer, and Peter Dalsgaard. 2020. Evaluating Creativity Support Tools in HCI Research. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) (DIS '20). Association for Computing Machinery, New York, NY, USA, 457–476. <https://doi.org/10.1145/3357236.3395474>
- [53] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking. *Institute for Software Research* 1, 816 (2005), 1–15.
- [54] Michael L. Rivera, Jack Forman, Scott E. Hudson, and Lining Yao. 2020. Hydrogel-Textile Composites: Actuators for Shape-Changing Interfaces. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–9. <https://doi.org/10.1145/3334480.3382788>
- [55] Devin J Roach, Craig M Hamel, Conner K Dunn, Marshall V Johnson, Xiao Kuang, and H Jerry Qi. 2019. The m4 3D printer: A multi-material multi-method additive manufacturing platform for future 3D printed structures. *Additive Manufacturing* 29 (2019), 100819. <https://doi.org/10.1016/j.addma.2019.100819>
- [56] David Roedel, Shaowen Bardzell, and Jeffrey Bardzell. 2015. Sustainable Making? Balancing Optimism and Criticism in HCI Discourse. *ACM Trans. Comput.-Hum. Interact.* 22, 3, Article 15 (jun 2015), 27 pages. <https://doi.org/10.1145/2699742>
- [57] Sowmya Somanath, Ron Wakkary, Omid Ettehadi, Henry Lin, Armi Behzad, Jordan Eshpeter, and Doenja Oogies. 2022. Exploring the composite intentionality of 3D printers and makers in digital fabrication. *International Journal of Design* 16, 3 (2022), 77–95.
- [58] Katherine W Song, Aditi Maheshwari, Eric M Gallo, Andreea Danielescu, and Eric Paulos. 2022. Towards Decomposable Interactive Systems: Design of a Backyard-Degradable Wireless Heating Interface. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 100, 12 pages. <https://doi.org/10.1145/3491102.3502007>
- [59] Specklesystems. 2022. Less Janky Grasshopper Components. Retrieved Sept 3, 2022 from <https://speckle.systems/blog/async-gh/>
- [60] Blair Subbaraman and Nadya Peek. 2022. P5.Fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment. In *Designing Interactive Systems Conference* (Virtual Event, Australia) (DIS '22). Association for Computing Machinery, New York, NY, USA, 1148–1161. <https://doi.org/10.1145/3532106.3533496>
- [61] Hongtao Sun, Jian Zhu, Daniel Baumann, Lele Peng, Yuxi Xu, Imran Shakir, Yu Huang, and Xiangfeng Duan. 2019. Hierarchical 3D electrodes for electrochemical energy storage. *Nature Reviews Materials* 4, 1 (2019), 45–60. <https://doi.org/10.1038/s41578-018-0069-9>
- [62] Lingyun Sun, Jiaji Li, Mingming Li, Yitao Fan, Yu Chen, Deyang Pan, Yue Yang, Junzhe Ji, Ye Tao, and Guanyun Wang. 2021. 3DP-ORI: Bridging-Printing Based Origami Fabrication Method with Modifiable Haptic Properties. In *The Adjunct Publication of the 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '21). Association for Computing Machinery, New York, NY, USA, 74–77. <https://doi.org/10.1145/3474349.3480233>
- [63] Haruki Takahashi and Jeeun Kim. 2019. 3D Printed Fabric: Techniques for Design and 3D Weaving Programmable Textiles. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 43–51. <https://doi.org/10.1145/3332165.3347896>
- [64] 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* (Seattle, WA, USA) (SCF '22). Association for Computing Machinery, New York, NY, USA, Article 8, 12 pages. <https://doi.org/10.1145/3559400.3561998>
- [65] Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2021. Tools, Tricks, and Hacks: Exploring Novel Digital Fabrication Workflows on Plotter/Twitter. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 594, 15 pages. <https://doi.org/10.1145/3411764.3445653>
- [66] Alexander D Valentine, Travis A Busbee, Boley John William, Jordan R Raney, Alex Chortos, Arda Kotikian, John Daniel Berrigan, Michael F Durstock, and Jennifer A Lewis. 2017. Hybrid 3D Printing of Soft Electronics. *Advanced Materials* 29, 40 (2017), 1703817. <https://doi.org/10.1002/adma.201703817>
- [67] Félix Vanneste, Olivier Goury, Jonas Martinez, Sylvain Lefebvre, Herve Delingette, and Christian Duriez. 2020. Anisotropic Soft Robots Based on 3D Printed Meso-Structured Materials: Design, Modeling by Homogenization and Simulation. *IEEE Robotics and Automation Letters* 5, 2 (2020), 2380–2386. <https://doi.org/10.1109/LRA.2020.2969926>
- [68] Eldy S. Lazaro Vasquez and Katia Vega. 2019. From Plastic to Biomaterials: Prototyping DIY Electronics with Mycelium. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers* (London, United Kingdom) (UbiComp/ISWC '19 Adjunct). Association for Computing Machinery, New York, NY, USA, 308–311. <https://doi.org/10.1145/3341162.3343808>
- [69] Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2020. Jubilee: An Extensible Machine for Multi-Tool Fabrication. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376425>
- [70] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. 2019. A-Line: 4D Printing Morphing Linear Composite Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300656>
- [71] Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 623–635. <https://doi.org/10.1145/3242587.3242625>
- [72] 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* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 5743–5752. <https://doi.org/10.1145/2858036.2858281>
- [73] Wen Wang, Lining Yao, Teng Zhang, Chin-Yi Cheng, Daniel Levine, and Hiroshi Ishii. 2017. Transformative Appetite: Shape-Changing Food Transforms from 2D to 3D by Water Interaction through Cooking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). Association for Computing Machinery, New York, NY, USA, 6123–6132. <https://doi.org/10.1145/3025453.3026019>
- [74] Malgorzata A Zboinska and Delia Dumitrescu. 2021. On the aesthetic significance of imprecision in computational design: Exploring expressive features of imprecision in four digital fabrication approaches. *International Journal of Architectural Computing* 19, 3 (2021), 250–272. <https://doi.org/10.1177/1478077120976493>