Demonstrating Dynamic Toolchains for Machine Control

Hannah Twigg-Smith
University of Washington
Seattle, WA, USA
htwigg@uw.edu

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

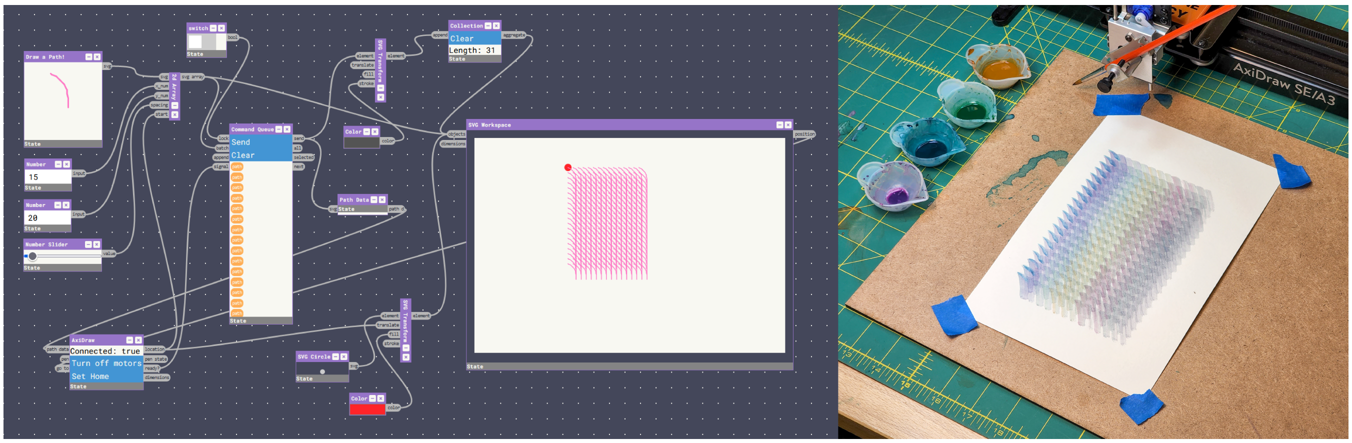

Figure 1: Dynamic toolchains can be used to interactively control machines for niche applications. Here, a dynamic toolchain for watercolor painting is used to step through a parametric toolpath created from user input.

ABSTRACT

Humans are increasingly able to work side-by-side with desktop-scale digital fabrication machines. However, much of the software for controlling these machines does not support live, interactive exploration of their capabilities. We present ***Dynamic Toolchains***, an extensible development framework for building parametric machine control interfaces from reusable modules. Toolchains are built and run in a live environment, removing the repetitive import and export bottleneck between software programs. This enables humans to easily explore how they can use machine precision to manipulate physical materials and achieve unique aesthetic outcomes. In this demonstration, we build a toolchain for computer-controlled watercolor painting and show how it facilitates rapid iteration on brush stroke patterns.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Software and its engineering** → **Integrated and visual development environments**.

KEYWORDS

toolpath design, plotting, watercolor, digital fabrication

SCF '22, October 26–28, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9872-5/22/10.
https://doi.org/10.1145/3559400.3565598

ACM Reference Format:
Hannah Twigg-Smith and Nadya Peek. 2022. Demonstrating Dynamic Toolchains for Machine Control. In *Symposium on Computational Fabrication (SCF '22), October 26–28, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3559400.3565598

1 INTRODUCTION

The CAD/CAM/CNC software pipeline traditionally used for digital fabrication workflows separates design and machine execution into distinct programs [Twigg-Smith et al. 2021]. This separation enforces a "static design practice" [Torres and Paulos 2015], where a bottleneck imposed by static interchange files prevents machine users from easily iterating on machine behavior. Every program includes decision points that influence how a machine will execute a fabrication task, which ultimately determines the aesthetic qualities of the eventual outcome [Zboinska and Dumitrescu 2021]. The lack of feedback between CAD, CAM, and CNC software forces machine users to predict how design decisions made early in their workflows will translate to machine behavior, which can make aesthetic outcomes hard to anticipate and difficult to explore [Hudson et al. 2016; Kim et al. 2017; Norouzi et al. 2021].

For example, Figures 2 and 3 show how the path a machine will follow while painting with watercolor does not indicate the nuanced way paints will mix, layer, and pool. Under a CAD/CAM/CNC workflow, a machine user must engage in a lengthy trial-and-error process of tweaking designs, importing and exporting files, and repeatedly setting up jobs to explore how different toolpaths will translate to different aesthetic outcomes. Alternatively, developing custom software for this workflow would require extensive implementation time because there are few tools to support fabrication

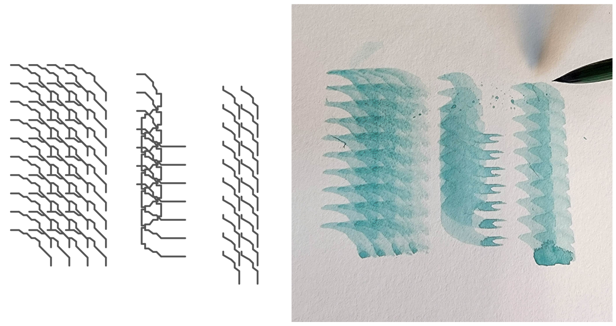

Figure 2: The toolpath for a series of brushstrokes looks very different from the eventual painting. Watercolor brushstrokes include pools of paint, overlapping layers of color, and bristle marks patterns. All of these are expressive dimensions that should be available to a machine user.

software development outside of the CAD/CAM/CNC paradigm. Seeking to re-couple human design intent with machine behavior, researchers have proposed new fabrication paradigms such as *interactive*, *lucid*, or *bidirectional* fabrication, where fabrication systems mediate human input and machine behavior [Kim 2017; Tian et al. 2019; Willis et al. 2010]. These contributions are often accompanied by promising proof-of-concept systems tailored to individual application domains.

We want to support further development of novel fabrication systems that enable unique collaborations between humans and machines. To this end, we contribute ***Dynamic Toolchains***, a extensible development framework for building alternative fabrication interfaces from live, interconnected, and reusable software tools. Our framework builds on prior work that has explored interactive machine control from programming environments [Fossdal et al. 2021; Subbaraman and Peek 2022]. Dynamic toolchains additionally provides support for custom graphical interface components, event-driven feedback between tool modules, and representations of real-world state. Our demonstration will show how dynamic toolchains can be used for interactive watercolor painting.

2 SYSTEM OVERVIEW

When designing the dynamic toolchains framework, we drew inspiration from parametric design tools (e.g. Grasshopper, Sverchok), component frameworks (e.g. React, Lit), and live programming environments. Our system has two main parts: a *live environment* for constructing and executing dynamic toolchains, and a *component framework* for developing the software tool modules which comprise toolchains. Tool modules resemble miniature, full-stack web applications with clearly defined APIs. Tools consist of three components:

- **JSON configuration file.** A short configuration file specifies the tool's basic information, including the tool name and description, input and output data ports, and internal state variables.
- **Python backend.** A tool's Python backend can include methods that handle human input, define custom behavior, manage internal state, and communicate with connected tools.
- **Web Component.** Optionally, tools can include a Web Component to serve as a graphical interface in the browser frontend. Web Components are a modern standard for encapsulating HTML, CSS, and JavaScript into a reusable custom element.

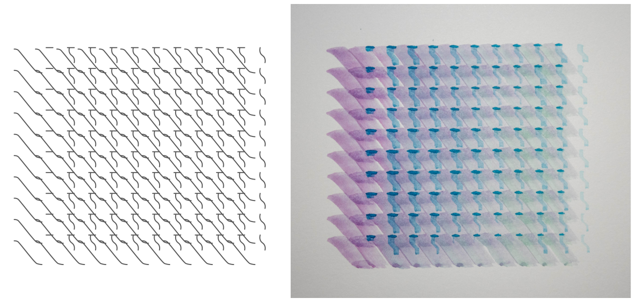

Figure 3: Left: a toolpath built interactively via a dynamic toolchain. Right: The resulting painting.

Our framework provides Python and JavaScript parent classes for tool modules. These have very few dependencies outside of standard language features and include lifecycle methods that tool modules can override for custom behavior Ultimately, we place very few restrictions on module implementation and functionality. We also provide a browser interface for constructing and manipulating toolchains. Our environment dynamically creates and maintains WebSocket connections between the front- and back-ends of each tool module.

2.1 Demonstration

Visitors to our demonstration will use a dynamic toolchain to explore brush stroke patterns for watercolor plotting, the interface for which can be seen in Figure 1. The painting toolchain consists of a number of interconnected modules, which can be broken into four categories: AxiDraw communication, visualization, parametric toolpath generation, and event planning. By adjusting module parameters in our live toolchain authoring environment, users of the system can easily iterate on geometry and quickly see the results from the machine. Visitors to the demo will create toolpaths by drawing with a stylus or uploading their own SVG input files. They can preview the resulting toolpaths, send them to the machine for testing, and create machine-painted artworks to take home.

3 CONCLUSION

We demonstrate *dynamic toolchains*, an extensible framework for building live digital fabrication machine control interfaces from modular interconnected components. Toolchains are authored in a browser-based environment by connecting modules with various functionalities, such as designing paths, creating machine commands, or sending commands to a machine. In this demonstration, users can modify the base path of a plotting toolpath to explore the different aesthetic outcomes of watercolor brushstrokes.

REFERENCES

Frikk Fossdal, Rogardt Heldal, and Nadya Peek. 2021. Interactive Digital Fabrication Machine Control Directly Within a CAD Environment. In *Symposium on Computational Fabrication (SCF '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3485114.3485120

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 (CHI '16)*. Association for Computing Machinery, San Jose, California, USA, 384–396. https://doi.org/10.1145/2858036.2858266

Jeeeun Kim. 2017. Shall We Fabricate? Collaborative, Bidirectional, Incremental Fabrication. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 83–86. https://doi.org/10.1145/3131785.3131844

Jeeeun Kim, Haruki Takahashi, Homei Miyashita, Michelle Annett, and Tom Yeh. 2017. Machines as Co-Designers: A Fiction on the Future of Human-Fabrication Machine Interaction. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 790–805. https://doi.org/10.1145/3027063.3052763

Behnaz Norouzi, Marianne Kinnula, and Netta Iivari. 2021. Making Sense of 3D Modelling and 3D Printing Activities of Young People: A Nexus Analytic Inquiry. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3411764.3445139

Blair Subbaraman and Nadya Peek. 2022. P5.Fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment. In *Designing Interactive Systems Conference (DIS '22)*. Association for Computing Machinery, New York, NY, USA, 1148–1161. https://doi.org/10.1145/3532106.3533496

Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New Orleans, LA, USA, 713–725. https://doi.org/10.1145/3332165.3347918

Cesar Torres and Eric Paulos. 2015. MetaMorphe: Designing Expressive 3D Models for Digital Fabrication. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (C&C '15)*. Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/2757226.2757235

Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2021. Tools, Tricks, and Hacks: Exploring Novel Digital Fabrication Workflows on #PlotterTwitter. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3411764.3445653

Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2010. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. Association for Computing Machinery, New York, NY, USA, 69–72. https://doi.org/10.1145/1935701.1935716

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 (Sept. 2021), 250–272. https://doi.org/10.1177/1478077120976493