UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Emergent Expression: Distributing Human-Nonhuman Creative Agencies in Computational Art Practices

Permalink

https://escholarship.org/uc/item/3zb6d2t4

ISBN

9798297664401

Author

Toka, Mert

Publication Date

2025-09-11

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Santa Barbara

Emergent Expression: Distributing Human-Nonhuman Creative Agencies in Computational Art Practices

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Media Arts & Technology

by

Mert Toka

Committee in charge:

Professor JoAnn Kuchera-Morin, Co-chair
Professor George Legrady, Co-chair
Professor Jennifer Jacobs
Professor Marko Peljhan

September 2025

Professor Jennifer Jacobs

Professor Marko Peljhan

Professor JoAnn Kuchera-Morin, Co-Chair

Professor George Legrady, Co-Chair

September 2025

Emergent Expression: Distributing Human-Nonhuman Creative Agencies in Computational Art Practices

Copyright © 2025

by

Mert Toka

to my niece, *Eliz Koç*

Acknowledgements

This research would not have been possible without the support of my advisors, collaborators, colleagues, and friends!

I would like to thank my advisors, JoAnn Kuchera-Morin and George Legrady, for their unwavering patience, guidance, and support throughout my time in MAT. I will be forever grateful for your encouragement and mentorship, for making me a better artist, and a better person altogether. I also thank my committee member, Jennifer Jacobs, for her invaluable suggestions and feedback throughout my research in ECL. I am grateful for your time, effort, and support, and for helping me unlock whole new sets of skills and perspectives, and for making me a better researcher. I also extend my gratitude to Marko Peljhan, for your encouragement and support. At times, talking to you helped me much more than you could imagine!

I would like to thank my brilliant collaborators Avi Farber, Alejandro Aponte, Ashley Del Valle, Camila Friedman-Gerlicz, Devon Frost, Diarmid Flatley, Eun-ha Paek, Iason Paterakis, Nefeli Manoudaki, Pilar Wiley, Raina Lee, and Sam Bourgault for their contributions to my research and teaching me so much! I would also like to extend my sincere gratitude to faculty members at Sabancı University, who taught me so much, inspired me to follow my convictions, and kindled the passion to bravely pursue this interdisciplinary research: Selçuk Artut, Selim Balcısoy, and Elif Ayiter.

I wish to extend my sincerest gratitude towards my colleagues, collaborators, and friends at MAT. None of this journey would be the same without your support and friendship: Sahar, Nefeli, Iason, Sam, Devon, Ana, Ashley, Alejandro, Stejara, Pau, Anshul, Anzu, Xindy, Aaron, Mengyu, You-Jin, Emilie, Weihao, Diarmid, Emma, Jazer, Gustavo. You are the best part of my experience in MAT! Similarly, I'm forever grateful for being lucky enough to find a second Turkish family halfway across the world; I'm

deeply grateful for your support and cherish everything we have done together: Asutay, Furkan, Berkay, Zeki, İlayda, Oya, Anıl, Umut, Beyza, Fatih, Taylan. All of you made me feel secure, comfortable, and at peace with myself in this foreign land! And same goes to my oldest friends, thank you so much for your unconditional love and support, even from vast distances: Merve, Cansın, Alicem, Dilara, Demet, Mert, Can, Ceren, Gülşah, Hande, Öçbe, Ozan, Ecem, Muratcan, Çağrı, Doruk, Bengi.

Thank you to my family members, Mürüvet, Gürer, Merve, Gökhan, for their unconditional love and support across the globe. Particularly Eliz, my little niece–I dedicate this research to you in the hopes that you'll never lose the passion for making art and expressing yourself unapologetically!

Curriculum Vitæ

Mert Toka

Education

2025 (expected)	Ph.D. in Media Arts & Technology, UC Santa Barbara, CA, USA
2018	M.S. in Media Arts & Technology, UC Santa Barbara, CA, USA
2015	B.S. in Computer Science & Engineering, Sabancı University, Istanbul, Türkiye

Publications:

Toka*, Manoudaki*, Paterakis* & Flatley. *Simulacra Naturae: Generative Ecosystem driven by Agent-Based Simulations and Brain Organoid Collective Intelligence.* **IEEE VISAP 2025**, Vienna, Austria. arXiv:2509.02924

Toka*, Frost*, Bourgault, Farber, Friedman-Gerlicz, Lee, Paek, Wiley & Jacobs. *Practice-driven Software Development: A Collaborative Method for Digital Fabrication Systems Research in a Residency Program.* **DIS 2024**, Copenhagen, Denmark. 10.1145/3643834.3661522

Toka. *Crafting the Computational: Artistic Production, Generative Systems, and Digital Fabrication.* **DIS 2024 Companion**, Doctoral Consortium, Copenhagen, Denmark. 10.1145/3656156.3665122

Del Valle, **Toka** & Jacobs. *Engaging Young People in the Expressive Opportunities of Digital Fabrication Through Craft-Oriented CAM-Based Design.* **DIS 2024**, Copenhagen, Denmark. 10.1145/3643834.3660693

Toka*, Bourgault*, Friedman-Gerlicz & Jacobs. *An Adaptable Workflow for Manual-Computational Ceramic Surface Ornamentation.* **UIST 2023**, San Francisco, CA, USA. 10.1145/3586183.3606726

Del Valle*, **Toka***, Aponte & Jacobs. *PunchPrint: Creating Composite Fiber-Filament Craft Artifacts by Integrating Punch Needle Embroidery and 3D Printing.* **CHI 2023**, Hamburg, Germany. 10.1145/3544548.3581298

Dodge, **Toka** & Bae. *DynamoVis 1.0: an exploratory data visualization software for mapping movement in relation to internal and external factors.* **Mov Ecol 9, 55, 2021**. 10.1186/s40462-021-00291-5

Toka. *Siren: General-purpose algorithmic composition and live-coding environment.* **M.S. Thesis, 2018**, UC Santa Barbara, USA. 38565

Toka, Ince & Baytas. *Siren: Interface for pattern languages.* **NIME 2018**, Blacksburg, VA, USA. 10.5281/zenodo.1302677

Ince & **Toka**. *Siren: Hierarchical composition interface.* **ICMC 2017**, Shanghai, China. 2027/spo.bbp2372.2017.003

Kaya, **Toka**, Bayrak, Bozkaya & Balcısoy. *Incorporating tabletop visual analytics into the decision-making process: a case study of retail banking.* **IEEE Vis 2015**, business|vis|15 Workshop, Chicago, IL, USA.

Exhibitions & Demos:

Simulacra Naturae @ **IEEE VISAP** 2025, Atrium, Universität für Angewandte Kunst, Vienna, Austria.

Distributed Agency Trails @ **Deep Cuts** 2025, CNSI & SBCAST, Santa Barbara, CA.

Simulacra Naturae @ **Deep Cuts** 2025, CNSI & SBCAST, Santa Barbara, CA.

The Edge of Chaos @ **Deep Cuts** 2025, CNSI, Santa Barbara, CA.

3D-printed Ceramic Artifacts by ECL Team @ **Latentville** 2023, SBCAST, Santa Barbara, CA.

PunchPrint Fiber-Filament Craft Artifacts @ **CHI** 2023, Hamburg, Germany

The Edge of Chaos: Growing Patterns in Digital Petri Dish @ **Art of Science** 2022, UCSB Library, Santa Barbara, CA

The Edge of Chaos: Growing Patterns in Digital Petri Dish @ **syMADES** 2022, CNSI, Santa Barbara, CA

Morphogenic Clay Vessels @ **syMADES - Open Labs** 2022, CNSI, Santa Barbara, CA

The Edge of Chaos: Growing Patterns in Digital Petri Dish @ **Art of Science** 2022, SB Museum of Art, Santa Barbara, CA

The Edge of Chaos @ **EXP. Hybrid.Science.Experimentation** 2021, Barcelona, Online

Dynamic Confluence @ **Bricolage** 2020, UCSB, Online

GANesis @ **M.A.D.E.** 2019, CNSI, Santa Barbara, CA

Siren @ **NIME** 2018, Virginia Tech, Blacksburg, VA

Subtripping @ **Immerse(d)** 2018, Live House, Los Angeles, CA

Siren @ **Invisible Machine** 2018, SBCAST, Santa Barbara, CA

Polar Tasks @ **Re-habituation** 2017, CNSI, Santa Barbara, CA

GOOGLEJACK @ **ArtsIT** 2014, Minerva Palace, Istanbul, Türkiye

Shroom @ **Sparks** 2013, FASS Art Gallery, Istanbul, Türkiye

Abstract

Emergent Expression: Distributing Human-Nonhuman Creative Agencies in Computational Art Practices

by

Mert Toka

Since the 1950s, computer artists have used generative systems to explore creative agency in collaboration with autonomous processes that exhibit emergent behavior and self-organization. In the 1990s, artistic and scientific investigations into life-like computation introduced aesthetic frameworks inspired by biological growth and digital morphogenesis. More recently, advancements in materials and fabrication technologies have made translating generative processes into the physical world increasingly feasible.

This dissertation presents findings from my artistic practice, conceptualized as *emergent expression*, in the context of distributed creative agencies involving computational and material processes. These processes can display novel behaviors that influence the form in unexpected ways. The research encompasses both virtual and physical domains, investigating how generative systems can be effectively integrated into digital fabrication and craft workflows. Through practice-led research and collaborative projects, I develop computational complex systems informed by artificial life aesthetics, as well as new fabrication methodologies tailored for craft applications. These systems result in hybrid prototypes that demonstrate how generative form, material behavior, and human intervention can be combined into cohesive artworks.

This investigation centers on emergence as a creative construct, builds on my art practice-based research, and leads to the development of novel computational tools, techniques, and workflows that negotiate human creative agency.

Contents

Curriculum Vitae **vii**

Abstract **ix**

1 Introduction **1**
1.1 Motivations . . . 3
1.2 Problem Statements . . . 6
1.3 Contributions . . . 13
1.4 Attributions . . . 14
1.5 Outline . . . 15

2 Background **17**
2.1 Emergence as a Creative Framework . . . 18
2.1.1 Historical Foundations . . . 18
2.1.2 Emergence in Complexity Science . . . 20
2.1.3 Emergence in Creative Expressive Media . . . 23
2.2 Generative Systems in Digital Art . . . 26
2.2.1 A Brief History of Generative Art . . . 26
2.2.2 Evolutionary Aesthetics and Artificial Life . . . 29
2.2.3 Parameter Control Approaches . . . 33
2.3 Material Behavior and Digital Fabrication . . . 37
2.3.1 Material Practices and Digital Craft . . . 37
2.3.2 Traditional Ceramic Pipeline . . . 40
2.3.3 Computational Fabrication Systems and Workflows . . . 42
2.4 Distributed Creative Agencies . . . 44
2.4.1 Human-Computational Co-creation Models . . . 44
2.4.2 Human-Material Agency Frameworks . . . 46
2.4.3 Human-Computational-Material Systems . . . 49

3 Distributing Agency between Human and Computational Actors **52**
3.1 Methodology . . . 53

3.2 Case Study: The *Edge of Chaos* . . . 55
3.2.1 System Design and Implementation . . . 56
3.2.2 Strategies for Parameter Control . . . 63
3.3 Case Study: *Simulacra Naturae* . . . 65
3.3.1 Visual System Design and Integration . . . 66
3.3.2 Data-driven and Responsive Composition . . . 68
3.4 Results . . . 70
3.4.1 Parameter Space Characterization . . . 70
3.4.2 Emergent Outcomes by Algorithm . . . 72
3.4.3 Presentation Context . . . 77
3.5 Discussion . . . 79
3.5.1 GPU-driven Framework Permits Increased Emergence . . . 79
3.5.2 Parameter Exploration as a Creative Practice . . . 81
3.5.3 Limitations . . . 82

4 Distributing Agency between Human and Material Actors 85
4.1 Methodology . . . 86
4.2 Case Study: *ClayToolkit* . . . 90
4.2.1 System Design & Implementation . . . 91
4.2.2 Physical Production . . . 96
4.3 Case Study: *CeramWrap* . . . 99
4.3.1 Adaptable Workflow Development . . . 100
4.3.2 Digital System Implementation . . . 104
4.3.3 Physical Production . . . 108
4.4 Results . . . 110
4.4.1 Material Behavior Experiments . . . 110
4.4.2 Surface Detailing Approaches by System . . . 113
4.4.3 Parametric or Manual Geometries . . . 122
4.4.4 Presentation and Professional Contexts . . . 125
4.5 Discussion . . . 127
4.5.1 Practice-driven Software Development for Material Production . . . 127
4.5.2 How Fabrication Labor Mediates Software Production, and When Software Cannot Address Clay 3D Printers' Limitations . . . 131
4.5.3 Limitations . . . 136

5 Distributing Agency between Human, Computational, and Material Actors 139
5.1 Methodology . . . 140
5.2 Case Study: *Morphogenic Clay Vessels* . . . 142
5.2.1 System Design & Implementation . . . 142
5.2.2 Physical Production Workflow . . . 146
5.3 Case Study: *Distributed Agency Trails* . . . 151

5.3.1 Digital System Design . . . 151
5.3.2 Physical Production Workflow . . . 156
5.4 Results . . . 161
5.4.1 Material Glitches and Breakages . . . 161
5.4.2 Emergent Outcomes Afforded by Generative Systems . . . 164
5.4.3 Presentation Context . . . 166
5.5 Discussion . . . 168
5.5.1 Real-time Digital Interaction for Physical Production . . . 169
5.5.2 Tensions between Digital Precision Expectations and Material Realities . . . 170
5.5.3 Limitations . . . 171

6 Conclusion **173**
6.1 Discussion . . . 174
6.1.1 Theoretical Interpretation of Emergent Expression . . . 174
6.1.2 Agency, Unpredictability, and Control in Hybrid Practice . . . 176
6.1.3 Revisiting Research Questions . . . 179
6.1.4 Limitations . . . 180
6.2 Future Directions . . . 181
6.2.1 Iterating on *Simulacra Naturae* . . . 181
6.2.2 Exploring Other Material Domains . . . 188
6.2.3 Extending Emergent Expression to Artificial Intelligence . . . 189

Bibliography **191**

Chapter 1

Introduction

Computer science and art have been historically perceived as disparate disciplines. Early digital technologies emerged in environments such as military-industrial complexes and research centers that were beyond the reach of most artists [1]. Regardless, artists have experimented with and adopted the computational technologies as an expressive medium since the early days of general-purpose computing in the 1950s [2]. Computation has unlocked numerous novel ways of creating artworks, such as translating data between different media [3, p. 214] and constructing systems that iteratively apply simple rules to produce sophisticated forms and novel behaviors [4, p. 11]. Since then, the integration of computational concepts into art practices has revealed significant opportunities for expanding human creativity [5, p. 17] and for studying what I describe as *emergent expression*. Below, I briefly introduce this conceptualization of my art practice before elaborating it in the upcoming sections and chapters.

The first component, *emergent*, can be broadly described as novel types of behaviors in systems containing many interacting parts. It is an important property of *complex systems*, which are characterized by non-linear relationships between components leading to adaptation and emergence [6, p. 13]. Most natural and societal phenomena display

emergent behavior; some examples are the brain, insect colonies, social networks, economy, and governments [6, pp. 4–11][7, p. 221]. In computational art, emergence appears when artists use *generative processes* that produce forms and patterns that were not explicitly prescribed. These processes may or may not model complex systems. For instance, some artists create intricate geometric patterns by setting up symmetrical relationships [8], while others model biological complex systems to create artificial forms inspired by natural processes [9]. My research builds on the latter motivation: a fascination with simulating and visualizing emergent patterns that underlie natural and social systems.

And *expression* refers to the process of conveying ideas, emotions, and experiences through a creative practice that composes various media into a coherent piece. One perspective that helps study this expression is *creative agency*, the capacity to influence a creative output. My generative art practice and research on computational tool development revealed versions of this agency, and each version affects the workflows, qualities, and aesthetics of the expressive artwork. Within my screen-based art practice, I noticed I can offload some of my creative decision-making to the unfolding of computational processes. I refer to this ability of software agents to generate surprising forms as *computational agency*, granted by a deliberate transfer of part of an artist's intent to emergent computational tools and algorithms [10]. This co-creation extends to physical space through digital fabrication technologies, where computational processes meet material production. Within my computational fabrication research, I have encountered unexpected and surprising behavior originating from material properties and environment interactions. Adopting the established scholarship, I refer to generative affordances inherent in 'active' materials [11] that become visible in the tangible manifestations of objects as *material agency* [12, p. 22].

This dissertation investigates origins, transformations, and distribution of creative

agency in art practices that contain virtual and/or physical generative components. Therefore, **emergent expression** describes relational capacities and processes of human and nonhuman agencies that afford the creation of aesthetic media experiences (i.e. the perceptual qualities of an artwork resulting from these processes). To describe how aesthetic and expressive potential of virtual-physical media emerge from engagements between people, computation, and materials, I use the term *distributed creative agency.*

In this conceptual framework, the creation of a hybrid artwork starts with artists setting the overarching structure by defining systems and rules, and allowing other agencies to influence the content and unfolding of a piece. For instance, an artist creating artistic data visualizations might determine visualization's formal properties (i.e. colors, render style) and structure (i.e. dimension mapping), yet the dataset itself determines how this structure unfolds into a particular form with distinct visual content. Similarly, an artist creating physical sculptures might determine the formal elements (i.e. shape, size, colors, textures), yet tools, materials, and their interactions dictate how design intent is reflected in the particular details of the finished piece.

This dissertation primarily investigates these formal negations between human and nonhuman agencies described within the creative construct of *emergent expression.* It builds on my art practice-led research, extends to supporting practices of other artists, and leads to the development of novel computational tools, techniques, and workflows.

1.1 Motivations

This section outlines the personal and historical contexts that motivated this research. First, I convey the factors from my lived experience that shaped this research and my evolution as a practitioner-researcher. Particularly, three personal drivers have been central to *emergent expression*: (1) co-designing emergent aesthetics with gener-

ative computational systems, (2) extending emergent aesthetics into various expressive modalities, and (3) supporting human agency across emergent expressive systems. And second, I situate these components within a broader motivational context in media arts, theory, and human-computer interaction (HCI).

Personal Context

My background in computer graphics and data visualization [13, 14] has seeded an interest in creative computing and has further enabled a focused exploration of computational tools' expressive potential in interactive audiovisual software [15, 16, 17]. As I continued developing creativity support tools for other artists, I began integrating them into my art practice. My engineering background naturally steered me toward exploring generative computational designs more than manual design spaces, and I leveraged my media engineering skills to develop generative computational systems that co-designed with me. This art practice revealed three considerations:

The first consideration involved selecting the *design domain and aesthetics*. My exploration across generative design domains (e.g. tiling, pattern sequencing, data visualization) revealed relationships between the compositional qualities and aesthetic perception of generative outputs. Exploring mathematical frameworks to modulate order and complexity, I developed an appreciation for balancing order and complexity in generative outcomes and natural patterns. Consequently, the aesthetics of emergent designs became a central focus, leading me to investigate the dynamics of living systems for artistic expression.

Secondly, I explored various *modalities of expression* to embody these generative designs. Building on the insights from my previous tangible data visualization project, *Tangy* [14], I transition from screen-based media (e.g. digital animation, visual art,

virtual reality) to physical fabrication. The tangible interaction and screen-free handling of forms using touch sensation led to investigating computational fabrication and craft, revealing opportunities (e.g. material generativity) and challenges (e.g. tacit knowledge, material constraints) for understanding emergent expression across physical and digital making.

Finally, my exploration of emergent expression across generative domains and modalities highlighted workflow considerations concerning the *distribution of creative agency* between myself and computational systems. I developed control structures and workflows to balance my creative autonomy with that of generative processes. In physical modalities, creative agency was further distributed among the inherent behaviors of physical materials and processes. This prompted the development of computational tools that support artists' creative agency while respecting the unique qualities and constraints of each practice and medium.

Historical Context

Artists and researchers have been interested in emergence (novel behaviors arising from decentralized local interactions) to invite or study surprise in a given phenomenon [18, 6]. Various disciplines such as cybernetics, media theory, generative art theory, anthropology, architecture, new materialism, philosophy, and complexity science have investigated it with various motivations. Artists were primarily motivated by constructing aesthetic systems displaying emergent outcomes [9], and sciences attempt to shed light into mechanisms of emergence [19, 4]. My motivations build on these contexts: co-designing emergent aesthetics with computational systems, extending these aesthetics across media and modalities, and supporting human authorship in systems that also include nonhuman agencies. I present brief summary of these historical motivations here,

and articulate related histories and previous work within these disciplines in Chapter 2.

The appeal of emergent processes originates from their capacity to introduce elements of surprise, multiplicity, and co-creation by decentering deterministic authorship. Artists view these systems as dynamic collaborators that introduce uncertainty and variation in form-making [20, 21, 22]. In computational contexts, generative methods contain iterative rules, feedback, and open-ended evolution that shift emphasis from a single outcome to processes or possibility spaces [1, 23]. In material contexts, craftspeople engage with material behavior rather than suppress it and acknowledge materials' active role in shaping form [20, 11, 24].

In these contexts, theories relating to distributed authorship situate human maker in relation to nonhuman components [25, 26, 27]. Instead of defining deterministic forms, human creative role becomes setting up structures on which to distribute, orchestrate, and synthesize the emergent results of nonhuman processes into expressive outcomes. These histories motivate *emergent expression*: composing artworks by operationalizing relations among people, algorithms, and materials so that form and meaning unfold through the interactions. Practically, this involves setting system structures and rules, and allowing nonhuman agencies to shape portion of the design outcome. Conceptually, it treats emergence as a creative method for understanding and producing hybrid artworks.

1.2 Problem Statements

The objective of this research is to investigate the connections between computational generative and physical material systems. These relationships become most visible in the transitions between modalities and influences artist's agency in *creativity*, *aesthetics*, and *labor* domains.

In digital generative workflows, creative authorship often involves programming, it-

erative rule design, and refinement, where novelty emerges from exploring unfolding systems over time. In physical production workflows novelty often results from the tacit knowledge gained through sustained attention to material behavior. Additionally, digital and material systems produce different aesthetic qualities. Digital generative works emphasize algorithmic structure and efficiency, while material artifacts evoke tangible and embodied aesthetics perceived through qualities like weight, texture, and imperfection. Lastly, aspects of labor differ between digital and material domains. As David Pye notes, labor is linked to the medium's inherent risks and irreversibility [28, pp. 212–215]. Digital generative environments allow for navigating algorithmic structures with minimal material cost, and most digital operations are reversible and risk-free. In contrast, material workflows rely on physical and dexterous labor. The lack of an 'undo button', for example, exposes practitioners to a greater irreversibility.

In working with emergent expressive media, computational complex systems function as abstract engines and produces self-organizing forms through dynamic and iterative processes similar to biological development. Digital systems simulating abstracted natural processes such as aggregation, branching, or diffusion, are referred to as *digital morphogenesis*. This approach allows artists to shape evolving structures, curate algorithmic outcomes, and collaborate with computational agencies. This digital design approach has parallels in theories for material production, specifically what Tim Ingold describes as a form-generating, or *morphogenetic*, model of making. In this model, the form is not imposed upon an inert material, but grows through active engagement with material forces and flows. Although Ingold developed this concept to contrast it with form-giving (*hylomorphic*) models in physical making, its underlying principles of process-based generation, collaboration with nonhuman agencies through joining forces, and responsiveness to emergent conditions align closely with how digital artists use algorithmic systems that evolve form rather than dictate it.

Despite this generative richness, translating these digital forms into physical artifacts through computational fabrication presents challenges. Exporting computationally generated forms to fabrication machines without considering material affordances risks reverting to what Ingold critiques as a hylomorphic model. This is particularly salient in CAD workflows within digital fabrication, where geometry is often treated as a final design artifact rather than as something that continues to evolve in dialogue with materials. Even computationally 'grown' forms can be rendered inert if their physical production does not account for the affordances of fabrication and material agency.

This dissertation argues that both computational systems and physical materials can be viewed as *agents* in the making process, and each contributes to the emergence of form through their own internal dynamics. As Ingold writes, "[i]f organisms grow, so too do artifacts. And if artifacts are made, so too are organisms. What varies is the extent of human involvement in the generation of form: but this variation is one of degree, not kind" [24, p. 22]. To describe this interaction between computation, material, and human agents, I use the term *distributed creative agency* resulting in outcomes demonstrating components of emergent expression. Resulting tension between abstract system logic, embodied material response, and degree of human involvement is central to this dissertation and to understanding contemporary generative practices that operate within hybrid domains. The following three interdependent subproblems break down this larger challenge further:

Controlling the emergence and surprise in digital generative systems through system design and dynamics: When digital designers work in a deterministic mode, they exert full control over the design process and the outcome. For instance, 3D modeling software is intended to be used in a linear manner, where the user manipulates the components of the design digitally using direct manipulation tools. Another example is

parametric design, where the designer determines important parameters that govern the intended design and explores the parameters deterministically. Unless randomization is introduced, this workflow will result in the same outcome per set of parameter values. However, generative systems are able to produce surprising, often hard to control, forms that are not prescribed by the system's designer.

One of the challenges of working with these systems is the extent to which the human artist has control over the outcome. Since these emergent forms heavily depend on the system design, initial parameters, and how it evolves and unfolds over time, artists need to balance the spatiotemporal aspects of the system's evolution, requiring a deep understanding of the system's behavior through many iterations of trial and error. Unless the artistic goal is to create primarily generative works (e.g. Andy Lomas, Jared Tarbell, Anders Hoff, and Refik Anadol), this lack of total control over the system state and evolution may present a challenge to artistic workflows. Therefore, most artists rely on using generative systems to complement their deterministic art practice as compositional elements rather than adopting a completely generative approach (e.g. Max Cooper, Sougwen Chung, and Casey Reas).

Fabricating emergent computational forms as physical objects using 'active' materials: Digital platforms allow artists to create arbitrarily complex forms, including those impossible in the physical world, but translating these digital geometries into physical artifacts involves a range of technical and material challenges and constraints. These constraints include the fabricability of digitally-based design using low-volume manufacturing, and the differences emerging when translating digital toolpaths into physical outcomes using the workflows and behaviors of active materials.

Mass manufacturing in industrial contexts might set up elaborate and expensive production chains to produce many objects imaginable, however, computational fabrication

technologies like 3D printers in low-volume production contexts have more limitations in the types of geometries that can be translated into physical reality. Architecture, design and computational fabrication researchers have explored the possibilities of this intersection through generative design, which aims to optimize specific metrics like weight, tension, or strain in the final physical artifact [29, 30]. While generative design enables the creation of objects with specific physical properties, it offers limited opportunities for emergent expressivity for the discovery and collaboration with material agents.

Engineered materials like PLA and ABS offer uniform material properties and behavior, and they are ubiquitous in 3D printing due to their flexibility and predictability. In contrast, natural materials like clay, glass, or wood have their own 'active' material state, as they may shrink, crack, or deform unpredictably. Their material qualities can only be discovered through a sustained embodied interaction and by gaining *tacit knowledge* about their material behavior under environmental factors like stress and heat. Similarly, it is possible to 'misuse' the engineered materials in a way that they are 'activated', resulting in more unpredictable and surprising material behavior. Activated uniform materials can act similar to the natural materials, and artists have used such techniques to create artifacts that were not possible with the intended use of them. For example, artist LIA creates sculptural works shown in Figure 2.8C with a custom software that controls the deposition path and machine behavior [31]. An example of this 'misuse' is to turn off an FDM printer's fans so that the extruded molten plastic cannot solidify fast enough and droops. These unpredictability in material characteristics determine the success of the making process and dictate possible types of fabricable objects. Therefore translating digitally generated forms into physical artifacts using computational fabrication workflows and 'active' materials can be particularly challenging.

In addition, craft practices and communities formed around natural materials function as an important component of the creative economy [32, 33, 34] and the most resilient

part of the world economy [35]. Technologies aiming to bring computational concepts into craft spaces must be considerate about the established practices of the craft. Top-down technology development without considering the existing craft knowledge and workflows might result in, at best, tools with little usability for real-world practitioners, or, at worst, harm creative communities that these technologies intend to support.

Orchestrating emergent results of distributed, more-than-human agencies: Traditional art practices primarily rely on a human artist's control and agency, with tools serving as extensions of their creative intent. However, when artists intentionally incorporate emergent processes into their workflow, they relinquish a portion of their agency to these processes. This transfer of control has been a subject of longstanding discussions regarding authorship in digital generative art [36, 37, 38, 39, 40], but becomes more intricate when physical material actors are introduced. The complexity around this agency distribution increases dramatically as we expand beyond purely digital processes to include material processes. The network of actors grows to encompass numerous nonhuman entities: materials with their inherent properties, fabrication hardware, tools, and environmental factors. These actors display physical uncertainties; for example, ceramicists talk about 'kiln gods' to describe the unpredictable nature of firing a clay object. Each of these actors contributes its own form of agency to the creative process, resulting in a complex web of interactions and influences.

When artists decide to distribute their agency across multiple actors, they place a unique responsibility on themselves. Rather than maintaining direct control over every aspect of the creation, they must adopt strategies for managing and orchestrating the emergent results of many interacting entities. In digital generative systems, practitioners continue to explore sophisticated control structures, such as finding the edge of chaos in parameter spaces [41, 42] to find a balance between novelty and coherence. When

working with physical systems, these control challenges are further increased by physical uncertainties surrounding the material processes. As much as it is challenging to come up with a high-level control structure over the material behavior, this material agency offers a source of generativity especially visible at its intersection with other actors used in computational fabrication technologies. The agencies of materials like clay, with their complex behavioral affordances from construction through firing, demand approaches that actively engage with agencies of all participating actors.

Both digital and material domains raise ongoing questions about how to orchestrate distributed creative processes effectively. It contains spaces of productive tension between generative processes and human authorship, and digital perfectionism and material unpredictability. To understand these tensions and how emergent computational systems and material processes can augment human creative agency, I guide this research with the following questions:

RQ1. What processes and procedures in computational and material domains enable designing with morphogenetic principles?

RQ2. How do the interactions between autonomous computational processes and material fabrication practices shape the distribution of creative agency among human and nonhuman actors?

RQ3. How can we design new computational systems, techniques or workflows to support emergent expressive artistic practices?

Box 1: *Definitions of key concepts.*

Agent/Actor. Human or nonhuman entities responsible for affecting or shaping a part of the art-making process. Derived from Latour [26].

Agency. The capacity of an actor to influence the creative outcome.

Control. The intentional and skillful manipulation of a part of the creative process leading to the final outcome.

Emergence. Surprising, unexpected, and unpredictable behaviors that are not explicitly prescribed but comes to being through interactions between multiple actors.

Morphogenesis. Principles of form generation in biological entities; mapped to the computational domain as 'digital morphogenesis'.

1.3 Contributions

This dissertation makes the following contributions:

1. Providing a conceptualization of my artistic practice as *emergent expression*, that help understand distributed creative agency in hybrid art practices operating at digital and physical spaces.

2. Documenting and analyzing practice-led research projects demonstrating the emergent implications of distributing human, computational, and material creative agencies.

    - The workflows for implementing and controlling computational complex systems for screen-based media to support my emergent art practice and co-designing with computational agencies.

- Novel material methodologies, techniques, and workflows for tangible media to support manual-computational art practices and co-designing with material agencies, particularly using clay.
- Integrative workflows of generative systems and physical production for hybrid media to support co-designing with computational and material agencies.

1.4 Attributions

Below I give attributions to my collaborators in the content I present throughout the dissertation:

- I developed the *Edge of Chaos* project presented in Chapter 3 in 2021 along with a Discord community initiated by artist Arsiliath [43], comprising engineers, researchers, and computational artists interested in artificial life and bioart. In addition to the feedback from this online community, I have gained constructive input from JoAnn Kuchera-Morin and George Legrady throughout the iteration phases. This work have been exhibited in group exhibitions in 2021 *EXP. Hybrid.Science.Experimentation* [44], in 2022 *SyMADES* [45], and in 2025 *Deep Cuts* [46]. The second part of Chapter 3 presents the visual components of *Simulacra Naturae* (and Section 6.2.1 presents all components), developed in collaboration with Nefeli Manoudaki, Iason Paterakis, and Diarmid Flatley. This work and has been exhibited at *Deep Cuts* in 2025 [47]. A part of the content in this section has been published [48] and exhibited at 2025 IEEE Visualization Art Program (VISAP).
- The content presented in Chapter 4 is the result of extended collaborations between HCI researchers and professional ceramic artists. In particular, Section 4.2 presents

the software I develop as a part of artist-in-residencies involving Avi Farber, Camila Friedman-Gerlicz, Eun-ha Paek, Raina Lee, Pilar Wiley, Sam Bourgault, Devon Frost, and Jennifer Jacobs. An expanded version of this work encompassing five software implemented throughout two residencies has been published and presented at *ACM DIS* 2024 [49]. Similarly, Section 4.3 discusses a material-informed manual-computational workflow for texturing ceramic objects, developed in collaboration with Sam Bourgault, Camila Friedman-Gerlicz, and Jennifer Jacobs. This work has been published and presented at *ACM UIST* 2023 [50].

- Two prototypes described in Chapter 5 were developed by myself, with feedback from Jennifer Jacobs and colleagues in Expressive Computation Lab.

1.5 Outline

This dissertation is structured into six chapters:

Chapter 2. Reviews foundational literature on complexity and emergence, generative art, material fabrication, and distributed creative agency to establish the conceptual and methodological frame for the dissertation.

Chapter 3. Investigates emergent phenomena in computational systems through two case studies, the *Edge of Chaos* and its data-driven deployment, *Simulacra Naturae* (digital components only), to examine how artists negotiate creative agency with artificial life systems. Material and cyber-physical aspects of *Simulacra Naturae* are outlined in Future Directions.

Chapter 4. Investigates material fabrication processes in manual-computational workflows through two case studies, *ClayToolkit* and *CeramWrap*, to discuss the role

of clay's agency in art practice and to highlight how integration of computational fabrication into craft workflows can enable novel ways of making.

Chapter 5. Presents case studies of emergence in hybrid digital-physical systems, *Morphogenic Clay Vessels* and *Distributed Agency Trails*, to focus on integration of nature-inspired generative algorithms in clay fabrication workflows.

Chapter 6. Synthesizes key findings, discusses hybrid creative systems and limitations, and outlines future work. It also presents the full prose of *Simulacra Naturae* along with planned development across material and digital domains.

Chapter 2

Background

In this chapter, I review the conceptual and technical literature that inform this dissertation's investigation into creative agency in hybrid digital-physical systems. This literature spans disciplines across complexity theory, generative systems in art and design, material-centered fabrication in HCI, and distributed agency theories. As Christiane Paul states, digital art is a transdisciplinary practice [1, p. 22], with the integration of physical production in digital practices, this categorization becomes more challenging. Despite their diverse origins, these works converge on a central inquiry: How do creative forms and meanings emerge when agency is distributed among humans, machines, algorithms, and materials, and how can we support these practices with computational methods and manual techniques? This chapter surveys the intellectual foundations that support this inquiry by examining the role of emergence as both a theoretical concept and a method for expression and creative production.

2.1 Emergence as a Creative Framework

Emergence offers a compelling conceptual framework for understanding how form, behavior, and meaning can emerge with little centralized control in both computational and material systems. Today, emergence is typically described with two characteristics: (1) a global behavior emerging from smaller local interactions, and (2) this global behavior cannot be traced back to the individual smaller parts [51]. In creative domains, emergence not only describes a complex behavior, but also transforms into a mechanism for aesthetic discovery. This section establishes foundations for the notion that emergence is more than a theoretical concept but a creative construct that helps artistic inquiry through understanding of agency and form. Here, I review the evolution of the concept of emergence, trace its philosophical roots to its formalization in complexity science, and lastly, explore how artists and designers have embraced it as a creative framework.

2.1.1 Historical Foundations

The concept of emergence has long intrigued scholars across sciences and humanities as a way to describe phenomena that cannot be reduced to the sum of their parts. In 1875, George Henry Lewes used the word to distinguish between 'resultant' and 'emergent' compounds in a chemical reaction [51]. He states that, although the effects of reactions are the 'resultant' of its components, we cannot always trace the steps of this process, and describes this effect as 'emergent'.

In 1920s, this definition has been adopted by the first wave of emergence studies. Researchers in philosophy, sciences and theology used *proto-emergence* as a reaction to reductionism [51], which aims to deconstruct a phenomenon into constituent parts and gain knowledge about the deconstructed concept by studying the parts independently [23]. This more-than-the-sum thinking was adopted across disciplines. For example, in the

early 20th century, Gestalt psychology embraced emergence and holism as core notions to understand perceptual grouping and figure-ground organization [52, p. 2].

This paradigm shift made holistic scholarly activities more prominent. In 1968, Ludwig von Bertalanffy published *General Systems Theory* with the aim of finding *isomorphisms*, or mathematical similarities, between the disparate models based on order, organization, and holism of different disciplines [53]. These isomorphisms provide a way of mapping similar concepts across different types of mathematical models. For example, Douglas Hofstadter extensively writes about isomorphs between formal systems and their interpretations in the cases of mathematics, music, and visual arts by emphasizing that such isomorphisms give way to explicit and implicit meaning of the systems [54]. In this sense, general systems theory does not aim to unite the diverse problems of different sciences; instead, it attempts to find general-purpose interpretations that can apply to the models of many disciplines [55].

Concurrently, Norbert Wiener's cybernetics framed organisms and machines as systems regulated by feedback and governed by information flows [56]. The Macy Conferences between 1946 and 1953 consolidated a transdisciplinary vocabulary of regulation, communication, and control across mathematics, engineering, biology, and the social sciences [55]. First-order cybernetics extended general systems thinking from structure to behavior by emphasizing self-regulation through feedback and information, anticipating later formalizations of emergence in complex systems. Second-order cybernetics foregrounded the role of the observer, situating designers and experimenters *within* the systems they steer.

These early conceptualizations of emergence marked a shift in how causality and form were understood as interacting systems over isolated ones. As scientific disciplines matured in the second half of 20th century, the limitations of reductionist frameworks became increasingly apparent, particularly when dealing with phenomena such as neu-

roscience, ecosystems, and social behavior. This paved the way to the second wave of emergence studies, *neo-emergence* [51], that contained more methodical investigations of complexity theory and applied computational models capable of simulation and visualization of interacting complex systems.

Emergence has been studied by sciences as technical phenomena and by artists as an aesthetic construct. This dissertation adopts a dual perspective by treating emergence both as an explanatory lens for the behaviors of complex systems and as a practical method for creating generative artworks in collaboration with nonhuman agencies. Sections 2.1.2 and 2.1.3 provides histories for scientific and creative accounts of emergence, respectively.

2.1.2 Emergence in Complexity Science

Complexity science emerged in the late 20th century as a transdisciplinary response to the challenges of modeling systems with nonlinear interactions, adaptive feedback, and multiscale behavior. One of the leading research institutions in the world, the Santa Fe Institute (SFI) of complexity, spearheaded formalization efforts of this approach and positioned emergence as a key explanatory mechanism in complex systems ranging from biology to computation [57]. In this second-wave framework, emergence is not simply a philosophical inquiry, but a measurable, often computationally demonstrable phenomenon that results from agent-based rules and decentralized interactions.

Melanie Mitchell defines *complex systems* as "large numbers of relatively simple entities organiz[ing] themselves, without the benefit of any central controller, into a collective whole that creates patterns, uses information, and [adapts via learning and evolution]" [6, p. 13]. The main feature of complex systems is non-linear interactions between large number of components operating without a centralized control mechanism. This non-

linearity causes feedback loops, resulting in adaptations to environmental changes and self-organized behavior. These new structures or functions, coupled with untraceable causal hierarchy, give rise to emergent behavior.

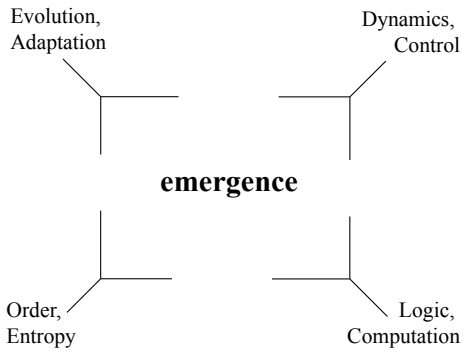


Figure 2.1: Four pillars of complexity science that support understanding *emergence* from different disciplines (adapted from Krakauer in [58, p. 25]).

Studying emergent behavior and formation of complexity science originated from distinct scientific disciplines. SFI director David Krakauer proposed an epistemological model shown in Figure 2.1 describing the four pillars of study that support technical developments around complexity and emergence: *Dynamics and control* (how systems change over time), *evolution and adaptation* (how systems adapt and reorganize), *entropy and order* (how systems store and transmit information), and *computation and logic* (how systems process and respond) [58, p. 25]. These pillars provide scientific methods for analyzing phenomena ranging from gene regulation to ant colonies, economies, and neural architectures [6, pp. 4–12].

A few important breakthroughs paved the way for the acceptance of emergent patterns in scientific disciplines (for a complete discourse, refer to [58]). Alan Turing pioneered the idea that simple mathematical equations could describe aspects of the biological world (particularly morphogenesis) through chemical interactions [59]. Boris Belousov discovered the Belousov-Zhabotinsky (BZ) reaction, which spontaneously oscillates in color

and self-organizes into waves and spirals [60], experimentally grounding Turing's ideas. Edward Norton Lorenz showed that even simple functions can behave unpredictably due to sensitivity to initial conditions and introduced chaos theory [61]. Robert May applied chaos theory to animal population equations, demonstrating how simple rules can yield unpredictability in nature [57, pp. 1244–1248]. Benoit Mandelbrot's fractal geometry demonstrated recursive self-similarity across scales, echoing coastlines and trees [57, pp. 934–940].

These pioneers connected emergent patterns and behaviors found in the natural world to mathematics and rule-based systems. As computing technologies became more powerful, computational researchers have experimented with many methods to study the mechanisms of emergence in natural world. A widely-studied computational complex system is the cellular automata (CA). Stephen Wolfram analyzed the behavior found in these systems and demonstrated a new method of scientific inquiry using computational systems [4]. Following Wolfram methods, Christopher Langton [41] and Matthew Cook [62] proved how CA supports universal computation. John Horton Conway's famous 'game of life' demonstrated novel computational structures (i.e. 'life-forms') emerging within a dynamic CA environment defined by rudimentary rules [63].

More recently, Rosas *et al.* introduces a computational blueprint of the functional architecture of nested complex systems [19]. This work highlights that macroscopic organisms can be 'causally-effective' in that understanding the microscopic behavior may not reveal insight into macroscopic behaviors (e.g. molecular structure of a falling object yields little information for its mechanics). More importantly, they find that examining the microscopic components can only give useful information about the higher-level behavior *when the emergent behavior is compromised* (i.e. system is broken). This finding resonates with Martin Heidegger's distinction of "ready-to-hand" and "present-at-hand" [64, pp. 95–107] discussed later in Section 2.4.2.

This rich scientific background offers creative professionals the vocabulary and a practical framework to construct, modulate, and analyze artistic systems that blur the boundaries between control and surprise. Emergence stands out as a form of computational material that artists and designers can influence but not entirely master. This productive tension between the structure and unfolding is the basis of emergent expression: an approach to making that leverages systemic behaviors to generate aesthetic outcomes.

2.1.3 Emergence in Creative Expressive Media

Artists and designers have long engaged with systems that exhibit unpredictable behavior arising from algorithmic variation or material resistance. Two early twentieth century art movements, Futurism and Dadaism, have been pivotal in building a bridge between technology and art [8, p. 17–20]. Futurists emphasized the artistic use of new technologies such as chronophotography, supported the dynamism of the modern world, and advocated for the artist-engineer hybrids capable of realizing artistic vision with technical knowledge. On the other hand, Dada movement rebelled against traditional notions of art with unconventional materials ('ready-mades') and various media experiments (collage and assemblage) [1]. More specifically, they prioritized chance and random processes over finished pieces. For example, poet and performance artist Tristan Tzara writes his poetry using the chance operations in cut-up technique (i.e. cutting words off of a newspaper and randomly pulling the next word for the poem) [8, p. 19].

Later in the century, Process Art extended these investigations of procedure and contingency by highlighting material behavior over fixed objects and argued for chance, gravity, and indeterminacy as form-giving forces [65]. For example, Robert Morris' 1970 piece *Untitled (Pink Felt)* drops cut industrial felt onto the gallery floor to exhibit its particular configuration emerging from material stiffness and gravity [66]. These art his-

tories prepared the ground for digital art's embrace of process, variability, and emergence over finished forms.

Philip Galanter develops a notion of *complexism* that reconciles differences between modernist science perspectives with postmodern cultures of humanities [67]. He juxtaposes the absolute values of modernity that reflects the Enlightenment ideologies with the relativistic views of postmodernity. He defines complexism as "the projection of the world view and attitude suggested by complexity science into the problem space of the arts and humanities". Complexism reframes artistic value not as a fixed endpoint but as the emergence of co-evolution between distributed agency networks. Other creativity scholars share similar views. Jason Hoelscher's concept of *art as information ecology* depicts art and artistic practice as an irresolvable dynamic system and difference-generating machine that resists reduction and closure [68]. He claims that art sustains the "most active, open, peculiar, and irresolvable mode" of information, yields increased complexity, and permits emergence across perceptual and cognitive levels.

A growing body of literature and creative practice make the aesthetic of emergence explicit in the context of generative art. Galanter, in a previous article, adapts the concept of algorithmic information complexity from Murray Gell-Mann, one of the founders of SFI, into the concept of "effective complexity", which is used to classify existing generative art algorithms capable of generating emergent forms [23]. Within the artificial life sub-field of generative art, artists cultivate computational systems such as CA and agent modeling to generate images, animations, and forms with life-like dynamics [37]. These practices treat systems as partners in making rather than as passive tools. The artist's task becomes designing generative conditions and orchestrating unfolding processes. Detailed histories and specific references to artists' works are discussed further in Section 2.2.

Beyond screen-based media, creative disciplines have pursued emergence through de-

sign and making. In architecture, O'Reilly *et al.*'s *Emergent Design* describe design methods that incorporate self-organizing and evolutionary processes over static form [69]. In computational craft contexts, Joshua Longo's *Emergent Making* articulates practice-driven material workflows that echo Ingold's morphogenetic stance: form as an outcome of engagement rather than imposition [70]. In HCI, Bennett *et al.* organized a CHI workshop on *Emergent Interaction* that invites complexity science approaches to analyze interactive systems, arguing for methods that account for non-linearity, feedback, and long-range dependence in human–technology assemblages [71]. Gaver *et al.* identifies challenges of emergent design in practice-based design research by contrasting emergence to intentionality in design research [72]. This dissertation aims to incorporate emergent design, making, and interaction in *emergent expression* by borrowing from digital design, material fabrication, and human-machine interaction.

Information aesthetics provided a complementary perspective on why emergent forms can be compelling. Early attempts to formalize aesthetic judgment, such as George David Birkhoff's *Aesthetic Measure*, relate perceived quality to a balance between order and complexity [73]. Following Birkhoff, Max Bense and Abraham Moles drew on Shannon's information theory to connect information, redundancy, and perception to study aesthetic experiences emerging from structured variation rather than from pure randomness or pure regularity [74, 75]. Frieder Nake describes these efforts as a 'heroic attempt' to describe how the field failed to capture human element and effectively reduced art works into probability distributions [76]. However, the field paved the way to contemporary generative art practices that bring these intuitions into a programmable space.

In creative practice, emergence offers not only a way to generate novel forms, but also a way to engage in dialogue with systems. It repositions the artist from a sole author to a participant within a network of constraints and possibilities. This reframing is central to this dissertation's investigation: understanding creativity not as an act of imposing form,

but as the navigation of an unfolding field of potential, shaped as much by machines, materials, and models as by human intent.

2.2 Generative Systems in Digital Art

Computational complex systems exhibit emergent and generative qualities, leading to unpredictable outcomes. I call this computational agency throughout this dissertation to discuss how a final form emerges from system setup and evolution rather than being imposed. In this section, I introduce a brief history of generative art, focus on evolutionary algorithms, present artistic works in artificial life, and review methods of human engagement with emergent systems. I do not, however, discuss the background of generative artificial intelligence (AI) models that recently became powerful for producing diverse media and artworks [77]. Since this dissertation does not delve into generative AI, I omit its background for brevity.

2.2.1 A Brief History of Generative Art

Artists have been using generative techniques as an expressive medium since the 1950s [2]. Early efforts in developing artistic computational tools have focused on simpler geometric generative forms that showcased the capabilities of computation of the time [1].

In its early days, the field was referred to as 'computer art' after the new and fascinating computing machines previously reserved for crunching numbers. Its transformation into a drawing platform was captivating and novel. In 1965, *Studiengalerie* in Stuttgart, Germany organized an exhibition titled 'Projekte generativer Aesthetik', and the term generative art was coined as a replacement for computer art [81]. A Siemens engineer at the time, Georg Nees, showed his computer-generated paintings [82]. In the same year, scientist Bela Julesz and engineer Michael Noll from Bell Labs showed

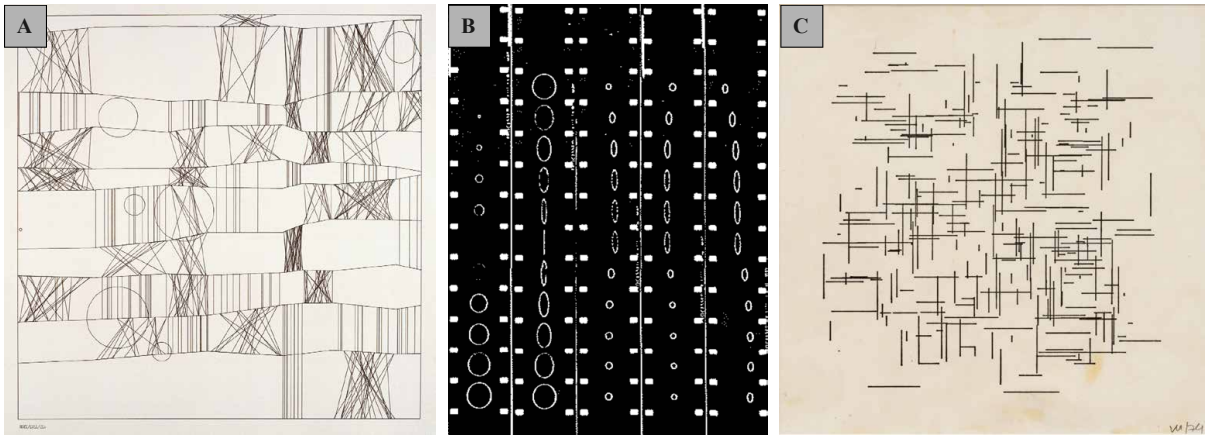


Figure 2.2: Early pioneering generative works of (A) Frieder Nake, *13/9/65 Nr. 2 ("Homage to Paul Klee")*, 1965 (from [78]), (B) John Whitney, *Permutations*, 1967 (from [79, p. 65]), and (C) Vera Molnar, *Molndrian, 74,066/13.36.22*, 1974 (from [80]).

'Computer-Generated Pictures' in *Howard Wise Gallery* in New York City, NY [76]. In 1968, *Cybernetic Serendipity* exhibition in London has gathered machine-art enthusiasts and announced this new exciting medium to the world [79]. The exhibition included computer art and other new forms of expression like electronic and kinetic art. In this exhibition, cybernetic researcher Gordon Pask demonstrated his conversational model in *Colloquy of Mobiles* (1968), where five hanging mobiles communicated via light and sound [83, p. 194]. Pask argued that adaptive, reactive machines form an "aesthetically potent environment" [84]. These pioneers adopted computing as a structural medium, collaborated with computational agencies for creating aesthetic content, and deeply influenced how artists understand authorship and agency in generative and computational practices [1, p. 9].

This computational agency that generates surprising behaviors and forms is the commonality between Margaret Boden's [37, p. 29] and Galanter's [23] definitions of generative art: autonomous processes displaying surprising behavior after iterative application of a rule set on the system state a sufficient number of times. These processes can be computational and discrete or physical and continuous. This definition emphasizes the

process, iteration, and emergence over static descriptions, aligning with this dissertation's focus on distributed creative agency.

These early developments resonate with art-historical currents that emphasize concept, event, formal instructions, and audience participation [1]. The Fluxus movement of the 1960s engaged formal instructions and 'found' elements. Building on Fluxus and conceptual art, digital technologies and interactive media further challenged traditional notions of the artwork, audience, and artist [1].

In addition to these early figures in computer-generated images, generative principles have been studied and applied in many disciplines, such as the music of Iannis Xenakis [85] and Brian Eno [86], the architecture of Michael Hansmeyer [87], and the user interfaces of Monolo Ide [88]. OULIPO extended this sensibility through constrained combinatorics in literature [1]. Boden and Ernest Edmonds provide an extensive taxonomy on recent practices in generative art in [37].

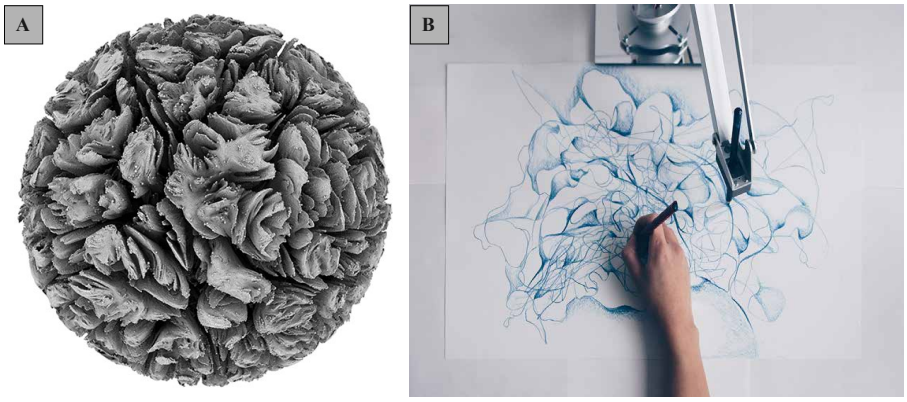


Figure 2.3: Images of two artists working with generative systems with different motivations. (A) Andy Lomas' 2014 project *Cellular Forms* is motivated by the emergent system development and exploration of output spaces. (B) Sougwen Chung's 2017 project *Drawing Operations* is motivated by emergent behavior design for human-robot collaboration.

Galanter notes that some artists are intrinsically motivated by the systems themselves, and treat system development and exploration as a creative and aesthetic act [23, p. 16]. For example, Andy Lomas creates biological generative systems that produce emergent

geometries [21] (Figure 2.3A). Other artists, like Sougwen Chung, use generative systems to produce hybrid human-machine aesthetics [89] (Figure 2.3B). Regardless, they use these processes as a collaborative partner that results in artistic forms that are not precisely designed by them [37, p. 29] and enables exploration of surprising content [90, p. 32]. Some artists are interested in this collaborative design paradigm, not merely what they generate, which makes the artist's role the designer of *hidden structures* that produce surprising *visible content*. Nake refers to these as *subfaces* and *surfaces*, respectively [74].

Some contemporary artists work primarily through generative systems; for example, Lomas' morphogenetic growth [21], Refik Anadol's data-driven environments [91], Jared Tarbell's algorithmic ornament [92], and Anders Hoff's procedural lineages [93]. Others integrate generative processes as compositional elements within broader artistic workflows; for example, Max Cooper's audiovisual performances [94], Sougwen Chung's human–machine drawings [89], and Casey Reas' software-based visual systems [95]. Artists often relinquish fine-grained control not as a deficit but as a strategy to access surprise, explore vast design spaces, and engage with emergence as an aesthetic resource.

After the 1990s, computers became powerful enough to simulate increasingly complex systems in real-time. This transition enabled artificial life to emerge as a subfield of generative art. Artificial life often involves evolutionary processes to develop forms resembling the form development in living entities like cells, plants, fungi, and animals. In the next subsection, I present the background on digital morphogenesis, evolutionary aesthetics, and artificial life.

2.2.2 Evolutionary Aesthetics and Artificial Life

The *aesthetic*, arguably the most subjective one, is one of the central motivations for their adoption in computational art. Here, aesthetics refers to the perceptible and

affective qualities of a particular *surface* (as Nake describes in [74]) capable of evoking an emotional, bodily, or cognitive response in its observer. For example, artists skillfully choose their aesthetic domain and control their medium in a way that the work evokes a sense of awe. Philosopher Alva Noë emphasizes the significance of such aesthetic engagement, describing it as a 'second-order, reorganizational activity' [96, pp. 29–32]. This second-order activity involves the act of exhibiting an artwork that investigates a certain subject matter, and the cognitive and affective interactions between the artwork and its audience create a discourse that ultimately transforms the work's subject matter into something new. Immanuel Kant sees aesthetic judgment as a paradoxical blend of subjectivity and universality, where personal feelings towards an artwork is supplemented with an expectation of universal evaluative norms [96, p. 201]. This contradiction generates a space of criticism and sustains the 'differential object' [68, p. 194] that creates the aesthetic value. Moles [75] and Wolfram [4, p. 549] have studied aesthetic perception using mathematical definitions of compression and Kolmogorov complexity.

Whether in information aesthetics [82, 75, 73], neuroaesthetics [97, pp. 41–44], computational aesthetics [40, 98], attempts at formalized and objective aesthetics studies encode perceptual principles into symbolic, mathematical, or algorithmic systems. While these fields contribute to structural understanding of visual and auditory systems, they risk reducing art to statistical distributions and fail to capture the cultural significance of fuzzy, human experiences [76]. This tension has fueled frameworks that treat aesthetic value as an adaptive, emergent, and evolutionary process.

Building on the philosophical and scientific accounts of evolutionary aesthetics, artificial life (a-life) operationalize emergent processes as generative engines of form and behavior. According to Langton, the field aimed to understand nature ('life-as-we-know-it') and to make systems that imagine new life ('life-as-it-might-be') [101]. This reimagination resonates with Aristotle, who proposed technē "completes what nature cannot"

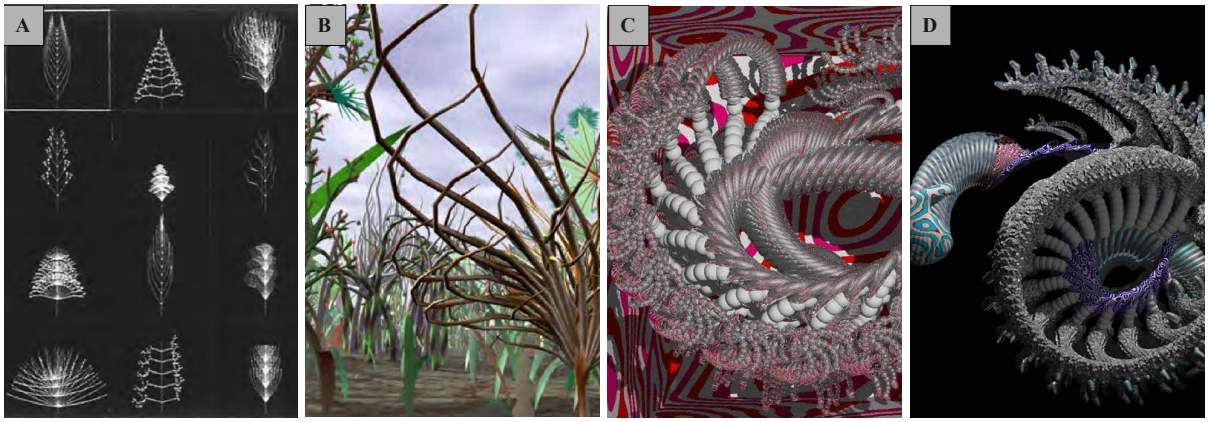


Figure 2.4: Stills from pioneering artificial life projects: (A–B) Karl Sims, *Artificial Evolution*, 1991 (from [99]); (D–E) William Latham's *Mutator*, late 1980s-early 1990s (from [100]).

by extending purposeful action beyond the organic realm [102].

Early computational models such as Turing's reaction-diffusion systems [59], Conway's Game of Life [63], and Craig Reynolds's Boids [103] demonstrated that simple local rules could produce rich, emergent patterns and behaviors. These systems use computation as a complexity-generating process, emphasizing interaction, adaptation, and feedback as the basis of organization. Recent a-life systems have extended beyond canonical algorithms toward increasingly complex, high-dimensional mathematical life-forms such as Jeff Jones' *Physarum transport networks* [104] and Bert Chan's *Lenia* [105, 106]. Leo Caussan's *Bibites* [107] and Christian Heinemann's *Artificial Life Environment (ALIEN)* [108] makes creating complex ecosystems containing thousands of entities accessible to public by presenting their systems in a video game format. Similarly, *Emergent Garden*'s browser-based *Life Engine* [109] allows a community of enthusiasts to create and discover artificial creatures. These platforms support open-ended evolution, decentralized adaptation, and emergent pattern formation at scales and speeds previously inaccessible.

By the late 1980s and early 1990s, increased computational power enabled a-life to

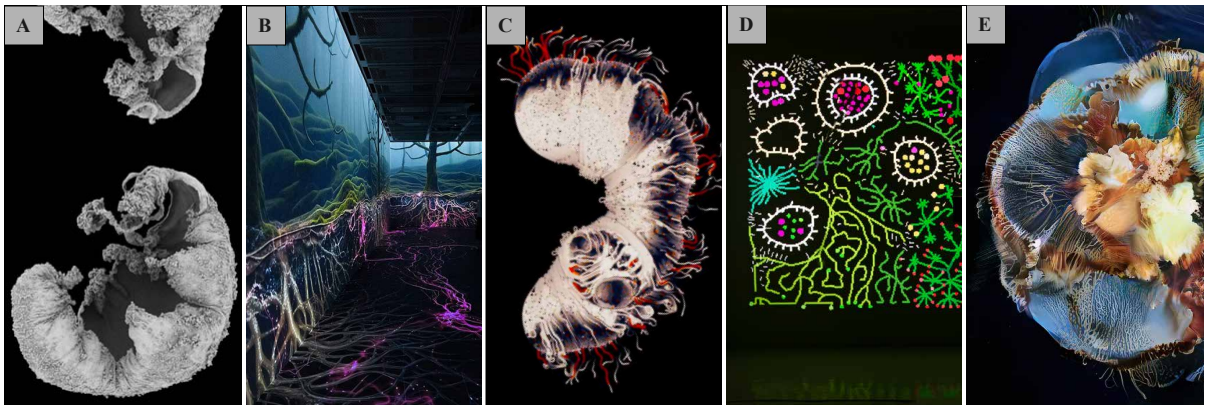


Figure 2.5: Images of recent artificial life projects: (A) Andy Lomas, *Constrained Forms*, 2017; (B) Haru Ji and Graham Wakefield, *Entanglement*, 2023–2025; (C) Arsiliath, *5*, 2021; (D) Spacefiller, *Living Walls*, 2018; (E) Sofia Crespo, *Neural Zoo*, 2018–2022.

develop as a distinct art discipline, drawing on genetic algorithms, swarm intelligence, and agent-based modeling to produce evolving forms. Canonical works include Karl Sims' *Artificial Evolution* using interactive selection to shape digital morphologies [99], and William Latham and Stephen Todd's *Mutator series* evolving biomorphic structures guided by human aesthetic choice [110] (Figure 2.4).

Many contemporary artists explore the aesthetic possibilities of life-like systems in virtual medium. Arsiliath's various untitled works [43] and Lomas' 2014 *Cellular Forms* generates morphogenesis via particle dynamics [21], while Ian Cheng's 2017 *Emissaries* generates narrative agents enacting a story among the open-ended chaos of the simulation [111]. *Spacefiller*'s 2018 *Living Walls* creates interactive ecosystem simulations with emergent ecosystems [112, 113]. Christa Sommerer and Laurent Mignonneau's 2015 *Living Poetry* transforms text and artificial creatures bidirectionally [114] and Sofia Crespo's 2022 *neural zoo* recombines diverse living species into reimagined natures [115]. Haru Ji and Graham Wakefield's 2025 *Entanglement* merges bio-inspired simulations with non-narrative storytelling to establish artificial ecosystems of forest and fungal networks [116]. Figure 2.5 shows a selection of these works side by side. Wu *et al.* surveyed the field of

a-life art and found that most works focus on the evolutionary dynamics of agent-based systems [9].

Artists have also expanded these methods into the physical space by integrating living organisms, robotics, and telepresence to foreground the agency of nonhuman actors (discussed more in Section 2.4). Eduardo Kac's 1999 *Genesis* encodes biblical text into bacterial DNA that participants mutate remotely, relocating authorship to a micro-ecological network [117]. Guy Ben-Ary's 2015 *CellF* cultivates neural stem cells into an 'external brain' that improvises on a modular synthesizer. Suzanne Anker's 2015 *Astroculture* and 2009 synthetic-biology project *E. chromi* further expand intelligence to plants and engineered bacteria, respectively, revealing how sensing, computation, and adaptation can operate across species boundaries [118, 110]. Contemporary bio-art transforms virtual life-like processes by framing living tissues, microbes, and plants as co-authors of creative systems.

2.2.3 Parameter Control Approaches

The behavior of emergent computational systems typically has a large number of executable pathways [39] and a vast amount of parameter combinations that shape the outcomes. Some portions of this parameter space result in more aesthetically appealing results than others. Artists frequently employ manual methods to explore and refine these spaces, particularly in short-form generative art, where curation and real-time feedback is central to the creative process. This contrasts with fully autonomous long-form generative art often used in blockchains, where parameter values must run unattended and evolve algorithmically without human intervention [119].

Manual Methods

A common manual approach is iterative trial-and-error with direct scoring and annotation of outcomes. Sterlin Crispin exemplifies this by running generative simulations under systematically varied parameters, capturing each result, and maintaining a spreadsheet with visual previews, parameter values, and subjective quality scores [120].

Tyler Hobbs describes a related but more exploratory process in his essays [121, 122], where he produces large grids of algorithmic 'sketches', each generated with different seeds or parameter sets. By visually scanning these collections, he develops intuition for how specific variables affect composition and texture. Hobbs emphasizes that this hands-on exploration builds 'a feel for the system' that cannot be replaced by analytical understanding alone, and that aesthetic learning often emerges from noticing unexpected outliers [123].

More specialized interfaces such as Lomas' *Species Explorer* helps explore systems with dozens of interdependent parameters by combining evolutionary operators with user scoring and lazy machine learning predictions [124]. Artists can work through phases while the system proposes new parameter sets based on prior evaluations. This makes the computer an assistant that guides the artist toward high-potential regions.

Similarly, *autonomX* provides a real-time compositional interface for complex systems [125]. Its direct-manipulation interface allows parameter manipulation without coding and sampling localized activity by directly drawing input/output regions on a visual grid. This interaction model privileges observation, listening, and gradual attunement to system behavior over abstract control, fostering a collaborative relationship between artist and algorithm.

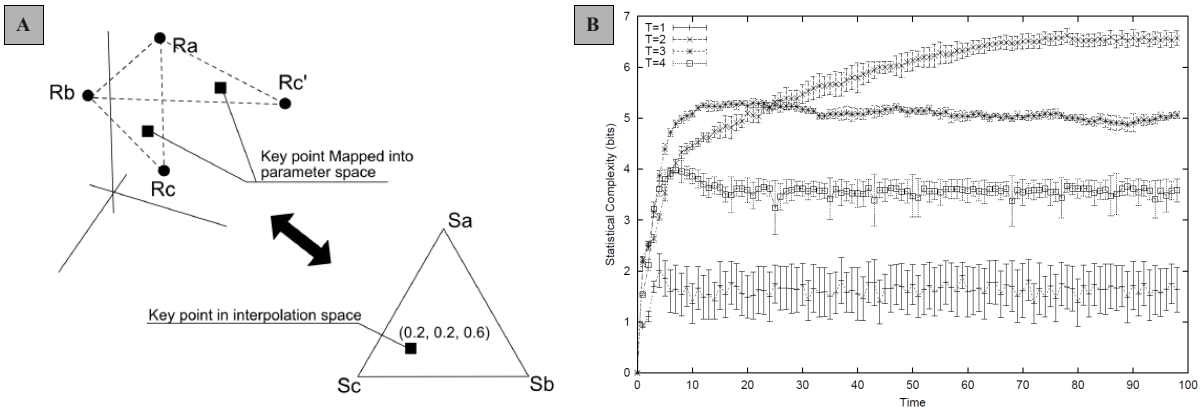


Figure 2.6: Diagrams depicting various algorithmic parameter control methods. (A) The barycentric coordinate abstractions remain the same while adjusting different part of the state space interface (from [126]). (B) Complexity over time for CCA with different thresholds T (from [127].

Computational Methods

One computational approach for managing generative algorithm parameter spaces is propagating higher-level descriptors down the system's conceptual space. This process does not analyze algorithm information content, but introduces a novel compositional interface for temporal control, reducing parameter space dimensionality and creating an interpolation space as a simplex geometry [126]. Artists can define a state space with reference points as simplex geometry vertices. In this geometry, higher-dimensional controls (e.g. barycentric coordinates from geometry surfaces in Figure 2.6A) efficiently reduce the number of parameters artists need to control.

Another approach is creating a general representation of the system's rules (a lookup table) and parameterizing it using information analysis. Langton applies this technique to elementary CA, defining a lambda parameter (λ) that controls its evolution [41]. This single, higher-level parameter ranges from zero to one and describes a CA starting from a fixed-point behavior, transitioning into periodic, complex, and chaotic as it increases (Wolfram classes I, II, IV, and III, respectively [4]). Only class IV CA displays complex

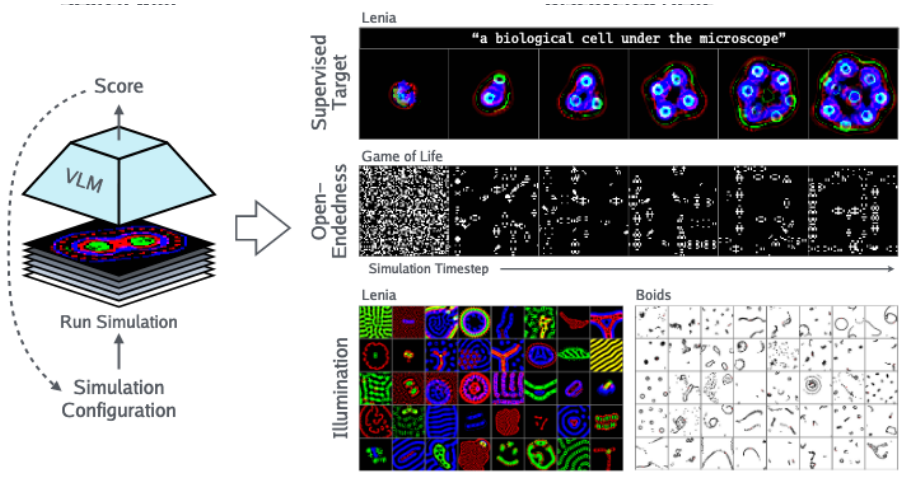


Figure 2.7: ASAL searching for interesting outcomes of artificial life simulations with vision-language models (from [129])

visual behavior with intricate structures at its edge of chaos.

Similarly, Shalizi *et al.* defines an information-theoretic measure for quantifying self-organization in cyclic cellular automata (CCA) [127]. Their method is highly general and easily applicable to other CA. The authors are against defining self-organization as reducing thermodynamical entropy, but in this work, they support the related concept of an increase in complexity. They measure the statistical complexity of many runs of CCA (Figure 2.6B) and point out that most visible complex patterns emerge when this statistical complexity is high.

Other researchers have also investigated the global information content of multi-agent systems. In contrast to the previous method, Parunak *et al.* quantifies the entropy reduction in self-organization in the stigmergy setup of ant simulations [128]. They use simple state partitioning and Monte Carlo simulations to show how coordination arises between the macro-level (agents) and micro-level (pheromones) processes.

More recently, researchers have used vision-enabled language models to identify regions of parameter spaces that yield interesting behavior. Kumar *et al.*'s method, ASAL (*Automated Search for Artificial Life*), finds simulations that exhibit life-like phenomena

by prompting a foundation model [129]. They also demonstrate open-ended search over large spaces. Because the method relies on generalizable models, it applies to diverse simulations such as Boids, Particle Life, Game of Life, Lenia, and CA.

These control mechanisms give artists an interface to collaborate with computational agencies that is discussed further in Section 2.4.1.

2.3 Material Behavior and Digital Fabrication

Computers and generative algorithms are not the only sources of surprise in hybrid artworks. The unpredictable behaviors of physical materials and fabrication environments can dictate the aesthetic qualities of a finished object. I call this *material agency* to discuss how a final form emerges rather than being imposed. For example, Hans Haacke's *Condensation Series* shown in Figure 2.8D exhibit unpredictable patterns of evaporating water vapor within a closed acrylic cube [38, p. 164]. In this section, I present material theories, specific clay workflows, and computational fabrication systems and workflows centering around craft-based production.

2.3.1 Material Practices and Digital Craft

Craft-based material production is among the oldest forms of making. As humans formed settled communities, division of labor enabled specialization in practices such as pottery and textiles, establishing enduring lineages of skill, technique, and material knowledge across regions including the Middle East [130] and Indigenous Americas [131]. These practices are grounded in intimate engagement with material behavior, where outcomes arise from iterative handling, environmental conditions, and tacit knowledge transmitted through making.

Debates distinguishing art and craft intensified in the late twentieth century, but

the boundary is historically blurry. Timea Tihanyi characterizes a persistent framing of craft as an ordered and systematic investigation, and art as a deliberate unsettling of order [20]. Importantly, this is not a simple opposition between control and chaos. In complexity science, chaos denotes sensitive dependence on initial conditions rather than an absence of structure. As Tihanyi notes, "we seem to be able to design chaos, but we are unable to design lack of order" [20, p. 101]. This perspective aligns with the broader arc of this chapter: both craft and art often work with structured indeterminacy, leveraging constraints to cultivate emergent outcomes.

Within HCI, researchers have articulated how craft practices resonate with digital fabrication research from material and practice aspects. Buechley *et al.* research material rheology (i.e. how materials deform and flow under stress) for finding new material recipes for digital fabrication [132, 133, 134]. Similarly, Bell *et al.* operate in bio-HCI and research bio-degradable materials and design pipelines [135, 136]. Bourgault shows that both traditional craft and computational workflows are organized around sequences of repeatable actions and proposes action-oriented interaction workflows that use craft actions as building blocks for hybrid manual–computational systems [137]. This framing helps connect embodied expertise with programmable processes, clarifying where human judgment, material agency, and algorithmic control meet in practice.

Complementing this, HCI and design research examine human-machine co-making, agency, and situated adaptation in hybrid craft contexts. Devendorf *et al.* reconfigure agency and control in hybrid fabrication, foregrounding negotiated authorship between makers and machines [138]. Andersen *et al.* articulates "digital crafts-machine-ship", explicitly positioning collaboration with machines as a craft competency [139]. Amy Cheatle and Steven Jackson analyze how craft and computation become entangled in real-world furniture practice [140], while Jacobs highlights cross-cultural hybrid practices that expand co-creation settings [141]. Together, these perspectives situate digital craft as a

material practice in which expertise, algorithms, and materials co-constitute outcomes.

Ceramics provides a salient case of such convergence. Around 2009, Dries Verbruggen and Claire Warnier's Studio Unfold and Jonathan Keep began developing extrusion-based clay 3D printing methods, combining open hardware and custom software [20]. By 2013, artists such as Keith Simpson explored subtractive CNC milling on clay slabs. In parallel, commercial companies like 3DPotter [142] and Vormvrij [143] expanded access to clay extrusion hardware, accelerating adoption in studios and schools. Building on this momentum, Keep published a comprehensive guide that documents practical parameters, workflows to show the opportunities and limitations of clay 3D printing [144].

The HCI literature on clay is relatively recent but growing. Researchers have developed computational approaches that embed data representations in printed ceramic artifacts [145, 146]. Others focused on surface finishing and post-processing. Dick *et al.* looked at laser-cutter-induced crackles in glazes to generate imagery on surfaces [147] and Zheng *et al.* explored the integration of interactive electronic circuits on glazed ceramic ware [148].

In this dissertation, I extend this landscape by developing manual-computational case studies that investigates how material behavior, toolpath construction, and human design intent interact to produce clay forms and surfaces. The projects explained primarily in Chapter 4 provides concrete case studies for examining material agency. Chapter 5 brings this discussion alongside computational generativity and studies how distributed creative agencies surface in hybrid computational practices. Next section situates clay 3D printing in the traditional ceramic pipeline and Section 2.3.3 details computational fabrication systems used to interface with physical production technologies in low-volume manufacturing settings.

2.3.2 Traditional Ceramic Pipeline

Traditional ceramic craft encompasses diverse skills, techniques, and materials that align with the dynamic material properties of clay. In this subsection, I present a summary of traditional ceramic manufacturing pipeline to present relevant background on where clay 3D printing fits into the established production methods. This background is important for understanding where emergent clay behavior in could arise regarding its form and surface details.

Ceramics manufacture is structured around multiple time-dependent steps that correspond with and are constrained by the material properties of the clay. It is important to stress that the pipeline presented below is ubiquitous for ceramic making and glazing; however, this summary is not exhaustive and variations exist. For example, many native potters skip the bisque-firing step altogether, or some ceramic artists exclusively collect wild clay and prepare it for pottery. This diversity of practice further demonstrates the need for adaptable approaches for integrating manual and computational steps in the development of new workflows and systems.

The firing temperatures are often referred to with *cone*s, and a 0 in the beginning of the number denotes lower temperature ranges. Most fire temperatures range from high-fire clay (Cone 10 – 1285 °C) to low-fire (Cone *0*10 – 900 °C). Cone 0 falls between 1119 °C to 1137 °C.

1. Preparing: Fabricating a clay object begins by preparing the raw material, clay, to a plastic state for moldability. Wedging by hand is the most common and accessible method, ensuring uniform consistency, improved plasticity, the removal of air bubbles, and particle alignment. Some craftspeople use industrial machinery like pug mills for churning and wedging clay bodies.

2. Form-making: Once the clay is wedged uniformly, the craftsperson chooses a ceramic design technique to create a form with wet clay. There are various alternatives such as wheel throwing, hand coiling, slab building, pinching, casting, 3D printing, or a combination of these techniques.

3. Drying: The craftsperson waits for a variable amount of time for the piece to firm up, ranging from hours to weeks depending on the scale and thickness of the object. During drying, water evaporates from the clay, and the object goes through distinct phases called *plastic / wet*, *leather hard*, and *bone dry* [149, 150].

4. Bisque firing: Once the piece is sufficiently dry, the craftsperson *bisque fire*s the piece in a kiln at lower temperatures (998 °C to 1063 °C/1820–1945 F) to remove all moisture from the clay and make it water-insoluble through partial vitrification.

5. Glazing: After the bisque fire, the ceramic piece is porous and ready to absorb glazes [151]. At this stage, craftsperson might add glazes on to decorate the surfaces of their object. Chemical composition of the glaze recipe dictate its final look after final fire. Some glazes produce shiny surface finish, and some creates crater-like details through chemical reactions.

6. Glaze firing: When all decorations are complete, the craftsperson *glaze fire*s the piece at a higher temperature (1031 °C to 1285 °C/1888–2345 F) to bring the clay to its final vitrified state. At this state, underglazes are chemically fused with the clay body, and glazes become glass-like [152]. After the glaze fire, the ceramic piece is finished and ready for use.

2.3.3 Computational Fabrication Systems and Workflows

Digital fabrication has been around in industrial settings since the 1950s with expensive machines and highly-specialized machine operators. As the patents of industrial applications entered in the public domain in the 2000s, desktop machines became available and this technology became more accessible to the general public. High precision and machine automation were initial selling points, but nowadays, many people use digital fabrication for rich sensory experiences, real-life interaction opportunities, high customization, and novel aesthetic outcomes [153].

Digital fabrication comprises two parts: digital design and physical production. Most digital workflows use a Computer-aided design (CAD) approach, requiring 3D modeling geometry in a CAD software and slicing it with another software that produces the machine toolpaths. After the initial setup and transferring the machine-readable file to the machine, the automated physical workflow starts and fabricates the object. This approach is great for most cases, but it can limit expressive goals. The computer-aided

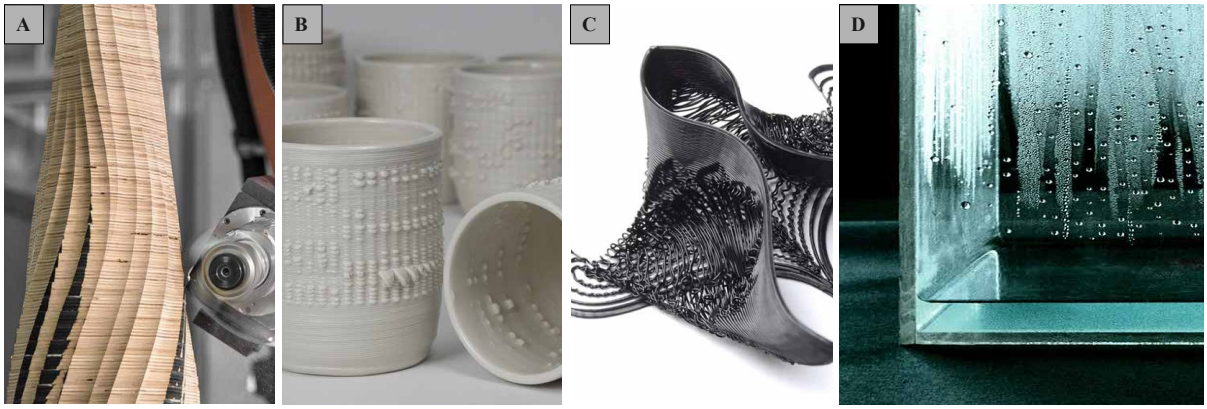


Figure 2.8: Examples of artists collaborating with various types of materials to reveal expressive opportunities: (A) ODK Design, *AESTUS*, 2019; (B) Slip Rabbit Studio, *Listening Cups*, 2018; (C) LIA, *Filament Sculptures*, 2014; (D) Hans Haacke, *Condensation Series*, 1963–1967.

manufacturing (CAM) approach, on the other hand, emphasizes creating machine toolpaths over creating a 3D model and slicing it. Using low-level machine instructions, it

directly interfaces with the machine and material, enabling the expressivity of digital fabrication technologies used with the unique agencies of the materials. Hirsch *et al.* interview the makers of nine design products (e.g. Figures 2.8A-B) and find that professional material workflows goes beyond CAD-based geometry and extends into low-level machine behaviors and material properties [154]. This subsection emphasizes CAM-based computational fabrication systems, workflows, and practices that expose toolpath construction as a creative medium, align machine control with material behavior, and support hybrid manual-computational decision-making in low-volume studio contexts.

Artists and researchers have used CAM pipelines as a design medium for novel fabrication techniques and material expression. Artist LIA uses a CAM approach to create the *Filament Sculpture* series (Figure 2.8C) using PLA [31]. Subbaraman *et al.* developed a p5 library that would make controlling the printer toolpath easier inside the p5js creative coding platform [155]. Verso integrates live parameter changes and styled toolpath visualization to support exploratory fabrication and mid-execution adjustment [156]. Dynamic Toolchains frames fabrication software as modular, reconfigurable toolchains tailored to practitioners' evolving workflows [157]. Vespidae proposes a programming framework for developing fabrication workflows and interfaces [158]. In the ceramic craft domain, Slabforge supports slab-based ceramics [159], CoilCAM enables action-oriented toolpath programming for coil-based printing [160], SketchPath brings gestural drawing into CAM for clay [161], and WeaveSlicer expands printable geometries by rethinking deposition patterns [162].

Across these systems, toolpaths become expressive mediums to negotiate flow, slump, shrinkage, adhesion, and post-processing. CAM-based and interactive control expose the temporal structure of making, enabling situated steering of outcomes. In my projects, I adopt toolpath design, inline parameterization, and ceramic-specific workflows to couple computational intent with clay's behavior and to foreground human judgment within

automated processes.

While these practices highlight key features of material behavior and workflow design, understanding how agency is distributed within them requires conceptual lenses introduced in the next section.

2.4 Distributed Creative Agencies

This section switches focus from the processes and workflows to theories and frameworks that discuss how creative agency is distributed among human, computational, and material actors. Hsueh *et al.* studies creativity often thrown around as a general term in HCI publications using an epistemological stance on the concept [163]. Bennett *et al.* reviews the concept of 'agency' and 'autonomy' used in thirty years of HCI research and finds that HCI has been using them as umbrella terms without clear agreement between researchers [164].

2.4.1 Human-Computational Co-creation Models

Artists, media theorists, and HCI researchers have investigated computational tools not only for their technical capabilities but because these systems enable *new forms of creative agency, collaboration, and aesthetics*. The motivation to engage with these tools originates from a desire to shape, challenge, or extend artistic workflows beyond the traditional techniques.

Computational agency in creative systems concerns with what the system can autonomously generate, transform, and evaluate, and how initiative is shared with human creators over time [165]. In HCI, agency is shaped by interaction techniques and represen-

tational choices that determine who can do what, when, and how [164]. In generative art, agency has been positioned as a goal and an analytic lens for creative systems [39, 10, 166]. Second-order cybernetics frames such co-creation as conversational control between human and system, as exemplified by Pask's work [84]. Generative processes reconfigure authorship, control, and responsibility over the creative process [74, 23, 76]. Computational agency is not a property of the machine alone, but rather the result of co-adaptive interaction. In this process, algorithmic affordances and constraints are continually shaped by human judgment and control.

Computational creativity scholars provide conceptual and practical foundations for co-creation. Canonical works outline how computational systems can explore, transform, and evaluate creative spaces [167, 90, 37, 168, 169]. Practitioner accounts and empirical studies show how artists work with generative systems in ways that share initiative and influence [36]. Recent methods align system outputs to individual aesthetic preferences [170] and consider complexity/agency relations in evolutionary art [171, 172].

Within HCI, creativity support technology (CST) research develops interactive tools that scaffold exploration, reflection, and production [173, 174, 175]. However, most CST evaluations remain short-term and outside expert practice [176], motivating calls for longitudinal, in-the-wild studies and closer ties between researchers and practitioners [177, 178, 179]. Although CST aligns naturally with social creativity and collaborative design perspectives [180], and with systems that help manage and compare alternatives during design exploration [181], skilled creative practices and artistic intents are underexplored in CST literature. Mappings of the field highlight a gap between CST research and tools used by creative professionals [182]. Research using the concept of 'power' revealed entangled relationships between the development of CSTs and the creative affordances of the software [183]. The abstractions and representations selected by the developers aimed to support and empower artists unavoidably exert a normative ground over the

artist's creative workflows.

Complementary to CST, artist-led tool development emphasizes creative agency. Artist Golan Levin advocate for open-source toolkits built by and for artists to serve their expressive needs because they support agency, experimentation, and accessibility [184]. Excellent examples are the creative coding environments *Processing* [185], initially developed by artists Casey Reas and Ben Fry, and *p5.js* [186], developed by Lauren Lee McCarthy. These efforts respond to mismatch between commercial creative tools and individual practice needs [183] and motivates artists to develop custom tools better reflecting their creative intentions [187].

In Chapter 3, I return to these co-creation models to discuss how I negotiated agency and control in my practice with the *Edge of Chaos* through developing hybrid simulations, generative ecosystems, and parameter control strategies.

2.4.2 Human-Material Agency Frameworks

Material agency concerns with how materials participate in shaping action, perception, and outcomes in making. Rather than treating matter as passive substrate, recent theories articulate how materials afford, resist, and co-direct practice. A foundational collection in archaeology and anthropology edited by Carl Knappett and Lambros Malafouris positions artifacts and materials as participants in social and cognitive activities [12].

Within the new materialism, Jane Bennett and Karen Barad advance strong ontological perspectives about matter's vitality and agential capacities. Bennett argues that matter has *vibrancy* that exerts force in political and everyday life and urges attention to, what Gilles Deleuze and Félix Guattari describe, *assemblages* in which human and nonhuman elements co-act [11]. Barad's *agential realism* frames phenomena as arising from *intra-action* rather than interaction to emphasize that agencies emerge through

material-discursive practices [188, pp. 139–141]. Similar to Barad's ontological view, recent perspectives from complexity science point to relational configurations between matter and information to explain problem-solving capacities of living systems [189].

While new materialist theories emphasize ontological agency or material forces, the Material Engagement approach [190] centers on how agency emerges with situated engagements and skillful practice. It focuses on enacting cognition through *brain–body–thing* relations, where thinking is materially extended and making is a cognitive process that unfolds through materials. Paul Louis March uses the Material Engagement approach to study how clay's uncertainties co-constitute agency in craft [191]. Extending Gilbert Simondon's notion of individuation [27], Kåre Stokholm Poulsgaard develops an account of *enactive individuation* in digital fabrication that shows how technics and materials co-shape sense-making over time [192]. Andrew Pickering describes such practice as a "dance of agency" where artists oscillate between active and passive phases containing interplay of resistance and adaptation of materials [165].

Ingold provides a detailed account of materials and making that is essential for understanding material agency in practice [193]. He argues that materials have histories and making is an ongoing correspondence with material flows rather than mere imposition of form [194]. As mentioned in Chapter 1, a central distinction is between *hylomorphic* (form-giving) and *morphogenetic* (form-generating) making [24, pp. 20–21]. In the hylomorphic model, a preconceived form is enforced upon matter, and in the morphogenetic model, the form emerges through guided negotiation with the material forces. This morphogenetic view is especially relevant for craft and computational fabrication that aim to use material dynamics in expressive setting.

In digital fabrication, this distinction manifests as contrasting models of practice. Replicator-style pipelines embed domain knowledge to automate production and minimize material engagement [195, p. 203], aligning with highly uniform, engineered mate-

rials and 'what-you-see-is-what-you-get' expectations [196]. By contrast, craft-centered workflows foreground negotiation with material behavior. As Tihanyi argues in the context of clay printing, the most interesting potential lies in the unexpected, resisting simple hylomorphic assumptions [20]. These stances frame material agency either as variability to be eliminated or as a collaborator to be cultivated.

Heidegger's phenomenology offers a complementary lens on material engagement through the distinction between *ready-to-hand* and *present-at-hand*. Tools, like a hammer, are primarily encountered in an ordinary world-view as ready-to-hand in the flow of skilled activity; however, when they fail to work in their intended use cases, they become present-at-hand as objects of detached meaning [64]. This shift illuminates how materials and tools disclose different agencies depending on engagement modes. Noë highlights that art's job is to take objects into present-at-hand and bring them into the contemplative view [96, pp. 199–200].

Simondon reframes technics and materials through his concept of individuation, examining the processes through which individuals and their social environments (*milieus*) co-emerge [27]. According to him, form and function of artifacts emerge through integration of internal and external constraints into a coherent whole, giving rise to *technical objects* whose efficacy depends on dynamic relationships between material, energetic, and functional regimes [197, 198]. This view emphasizes making activity as the guidance of emerging processes, aligning with morphogenetic accounts and informing contemporary reflections on design and fabrication [199, 200, 192]. Similarly, Bruno Latour's Actor Network Theory (ANT) provides a relational vocabulary in which materials and artifacts are *actants* whose capacities arise from networks of association [26, 201]. In ANT, material agency is not an intrinsic quality but enacted through heterogeneous assemblages. I present a discussion of the distributed and hybrid agencies of ANT in Section 2.4.3.

Chapter 4 presents case studies that applies material engagement to software devel-

opment and digital fabrication using clay's behavior and constraints.

2.4.3 Human-Computational-Material Systems

In emergent art and design practices, making involves heterogeneous groups of human, computational, and material actors whose capacities are co-constituted in practice. This section combines previous two sections to advance this dissertation's central stance: creative agency in hybrid systems is distributed across computational processes, material processes, and human intent and judgment. The question of how this creative agency is enacted and redistributed through the organization of tools, representations, and material engagements over time has been discussed in scholarly literature.

As mentioned in the previous section, Simondon's notion of individuation explains how technical objects and their associated milieus co-emerge and concretize through dynamic couplings [197, 27]. This view frames hybrid making as guidance of metastable computational and material processes that progressively concretize constraints. Latour's ANT complements this view with a descriptive vocabulary of generalizable actants and associations without "designat[ing] a domain of reality" [201, p. 72], using networks to explain how capacities arise in the configuration rather than in isolated components [26]. Resonating with Latour and Simondon, Lucy Suchman emphasizes the "inseparability of the human from the artifactual" and articulates a view of agency as distributed rather than solely residing in human or machine entities alone [202, p. 243].

Galanter's *complexism* situates generative art within complexity science, emphasizing bottom-up emergence, multiscale organization, and nonlinearity as aesthetic and methodological commitments [25]. Hoelscher extends this sensibility to the artworld by conceptualizing artworks and practices as information ecologies, where meaning and effect arise from densely interdependent flows among agents, media, and contexts [68].

Together these accounts support the view of hybrid agency as ecological: co-produced by algorithmic exploration, material behavior, and human interpretation.

Design approaches that prioritize the full life cycle of artifacts further underscore this ecology. Neri Oxman's "material ecology" proposes design that couples fabrication logics, material systems, and environmental relationships [203]. Work on unmaking and failure foregrounds how decay and breakdown contribute to creative possibility [204].

Artists working with generative methods in material practices often adopt stances that deliberately distribute creative agency. Lomas adapts morphogenetic simulations for 3D printing by encoding physical constraints in digital growth processes [205]. Oxman's lab develops biomaterials and fabrication techniques to create artifacts tuned to environmental and bodily contexts [203]. *Nervous System* uses highly-customizable generative design and low-volume material production to create nature-inspired artifacts and personalized jewelry [206]. *Urban Reef* uses computational to simulate objects containing micro-climates for various biological entities to grow on, fabricates these often with clay and deploy in a city environment [207].

These stances treat computation as a proposition and constraint engine, and material as a collaborator with tendencies to be harnessed rather than suppressed. They align with practice-driven accounts that emphasize emergence and negotiation in design and making [36, 72]. Evaluative attention shifts from ideal forms to process qualities, robustness under fabrication constraints, and experiential outcomes, consistent with complexity informed aesthetics [171, 172].

These exemplars demonstrate the value of hybrid generative-material workflows while also revealing accessibility constraints around the need for specialized facilities, multidisciplinary teams, and long development cycles. This motivates tool and practice research that lowers barriers by externalizing constraints, managing alternatives, and supporting in-the-wild iteration.

These works collectively shape the landscape on which this dissertation is situated. By interweaving concepts from emergence, computational generativity, material engagement, and distributed agency, I establish a foundation for exploring how hybrid creative systems challenge and expand notions of authorship, agency, and aesthetic control.

In the next three chapters, I will be discussing my research methodologies with the help of prototype projects. Chapter 3 discusses in what ways collaborating with computational emergence can produce novel aesthetics for screen-based media. Chapter 4 discusses how collaborating with the material agency of natural materials and expert craftspeople can produce a source of physical generativity in digital fabrication contexts. Lastly, Chapter 5 discusses how distributing human, computational, and material agencies can produce a hybrid, emergent expression.

Chapter 3

Distributing Agency between Human and Computational Actors

One of the primary components of emergent expression is the generative power of computational agents. These computational agents can display partial autonomy by producing forms and patterns that resemble morphogenetic development found in natural systems. In this setting, artists need to develop strategies for (1) designing a computational system that aligns with their design intent and (2) controlling the parameters and evolution of the system to balance artist's agency with that of the generative system.

In this chapter, I focus on how constructing and controlling computational complex systems in artistic workflows can yield nature-inspired forms for visual art. I first present the development and iterations of the *Edge of Chaos* prototype to demonstrate system construction, evolution, and real-time control. I then introduce *Simulacra Naturae* as a data-driven, large-scale case study that deploys the *Edge of Chaos* visual engine, and restrict discussion to its digital components. I present my findings regarding how these considerations enable balancing surprise and control in my art practice.

This chapter primarily addresses RQ1's computational component regarding the pro-

cesses and procedures resulting in morphogenetic form development. I employ custom systems designed around artificial life algorithms of cellular automata (elementary, cyclical) and agent-based systems (Boids, Physarum) as the primary morphogenetic processes. Developing automated and manual procedures to iteratively explore the parameter spaces of these processes enable co-designing with computational agents. These procedures contribute to RQ2 by describing the distributed creative agency component between human and computational actors. Similarly, the processes in this chapter contribute to RQ3 by detailing the computational system design of *Edge of Chaos* and its deployment in *Simulacra Naturae*.

3.1 Methodology

In this chapter, I use a practice-based research methodology [208] through implementation and artistic use of artificial life systems to examine their emergent aesthetics, self-organizing behaviors, and computational agency. Reiterating Langton, these algorithms offer artistic opportunities to study 'life-as-it-could-be' [101] and allowed me to discover new morphogenetic forms and behaviors. Following Linda Candy and Edmonds' methodology [208], this chapter situates the *Edge of Chaos* and its deployment in *Simulacra Naturae* as prototypes that (1) embeds practice into creative research inquiry, (2) reveals iterative experimentation and reflection, (3) demonstrates by making artifacts, and (4) engages the public by disseminating through exhibitions. The system implementation, control strategies, artifacts, and exhibitions act as the basis of contribution to knowledge of this case study.

The formative phases of the *Edge of Chaos* unfolded through cycles of ideation, implementation, exploration, and critique sessions spanning over eight months in 2020–2021. I've had implementation help and artistic feedback from an online community formed

by Arsiliath [43]. This community gathered artists and researchers who are interested in the aesthetics of artificial life simulations. GPU-powered system implementation in Unity [209] and strategies for real-time parameter-space exploration resulted in novel emergent aesthetics through a collaboration between my design intent and that of the generative systems.

Extending *Edge of Chaos* to a large-scale format in 2025, *Simulacra Naturae* deploys the visual system within a hybrid art installation developed in collaboration with Nefeli Manoudaki, Iason Paterakis, and Diarmid Flatley. The installation combines *Edge of Chaos* with collaborators' digital morphogenesis algorithms. The central driver for many components is pre-recorded human brain organoid activity from the Kosik Neurobiology Lab [210], and this dataset was used in my collaborators' previous installation [211]. I was not involved in collecting this dataset. This project uses it as an external signal to study dynamic parameter control and emergent behavior.

Throughout the exploration phases of the *Edge of Chaos* development, I have also had a chance to receive critique from my dissertation advisors; Kuchera-Morin is a distinguished electroacoustic multimedia composer and Legrady is a distinguished computational photographer and image maker. This expert critique revealed insights into compositional aspects of the outcomes, specifically the visual balance and temporal unfolding aspects of the outcomes. I remotely met with the experts weekly or biweekly over the course of two months, and we discussed the works in an unstructured show-and-tell format.

At the end of the eight month period, the project had produced hundreds of 2D and 3D animated recordings with various parameter configurations of four core algorithms (described below). I disseminated these recordings in a dedicated Instagram account, *ViscousVoid* [212], to collect public perception on my outcomes. This account acted as a portfolio diary on the actions I took to collaborate with computational agencies.

Following the formative phases of the project, I exhibited output artifacts in group exhibitions as a means to display novel aesthetics that communicate the research inquiry to the public. For *Edge of Chaos*, I used mixed output modalities such as digital or framed prints of still images, pre-recorded animated videos, real-time interactive visuals, and large-scale projection mapping. For *Simulacra Naturae*, we created a room-size environment containing various tropical plants resembling natural forests and two wall projections for the visual component. Section 3.4 elaborates how each output format demonstrates different aspects of computational agency in relation to emergent aesthetics:

- EXP. Hybrid.Science Festival, *Barcelona, (remote)*, 2021
- Art of Science Competition, *Santa Barbara, CA*, 2022
- SyMADES, *Santa Barbara, CA*, 2022
- DeepCuts, *Santa Barbara, CA*, 2025
- VISAP: Collective Care, *Vienna, Austria*, 2025
- Toka*, Manoudaki*, Paterakis* & Flatley. *Simulacra Naturae: Generative Ecosystem driven by Agent-Based Simulations and Brain Organoid Collective Intelligence.* In *2025 IEEE VIS Arts Program* (IEEE VISAP '25), Vienna, Austria [48]

3.2 Case Study: The *Edge of Chaos*

This project contains a set of generative computational systems examining the emergent aesthetic qualities of complex systems for visual media art. It constructs a virtual ecosystem in which various artificial species interact, grow, and evolve in response to their environments. These nature-inspired generative algorithms create interdependent relationships within the virtual ecosystem, display emergent behaviors, and result in self-organizing forms demonstrative of their computational agency. With this project, I investigate novel aesthetic possibilities, digital workflows for algorithm design, and strategies for creative output exploration.

The project's name was inspired by Langton's 1990 paper titled *Computation at the Edge of Chaos* [41]. This concept (or the *edge of criticality*) refers to a region in a system's behavior that transitions between order and disorder. Systems in this zone are not fully stable or fully chaotic but they exist in a phase transition state, exhibiting dynamic patterns, adaptive complexity, and longer transients. In this zone, small *quantitative* changes to system parameters can trigger rich and strong *qualitative* differences in the system output. For example, small temperature changes around 100°C can trigger molecular state changes between water to vapor. Langton shows that cellular automata (CA) tuned to this critical region can support complex computations and emergent structures. I adapted this term as my project name with the goal of reflecting exploration of emergent creative potentials of generative systems.

3.2.1 System Design and Implementation

I developed *Edge of Chaos* using Unity, C#, and HLSL. It contains four core algorithms, specifically elementary cellular automata (CA) [41], cyclical cellular automata (CCA) [213], agent-simulations with direct-communication [103] and with indirect communication (stigmergy) [104]. These four systems represent two classes of generative systems: The first, *grid-based systems*, are composed of discrete cells, and the whole space is updated at the same time based on rules for local cell interactions. In the second, *agent-based systems*, agents occupy a continuous space, and the simulation updates each agent's parameters based on the agent's communication method. In direct-communication, flocking agents update their parameters with respect to other agents' values, and in stigmergy communication they deposit and capture communication through their environment.

I utilized GPU-programming to implement simulation systems using compute shaders in HLSL. GPU-programming allows for massively parallel execution of instructions and

allows for simulating millions of component interactions without compromising real-time interactivity. In addition, being able to simulate more agents aligns with foundations of complexity; specifically the application of simple rules many times producing complex and emergent results. Massively parallel execution in GPUs is well suited for increased emergence and self-organization. All algorithms contain compute shader files containing the instructions for the GPU (core simulation), and C# files containing instructions for internal system instructions that run on the CPU (interface, data management).

Core Algorithms

Here, I briefly describe each core algorithm used in this project. I extended portions of the original algorithms for varied visual behavior.

1. Cellular Automata (CA):

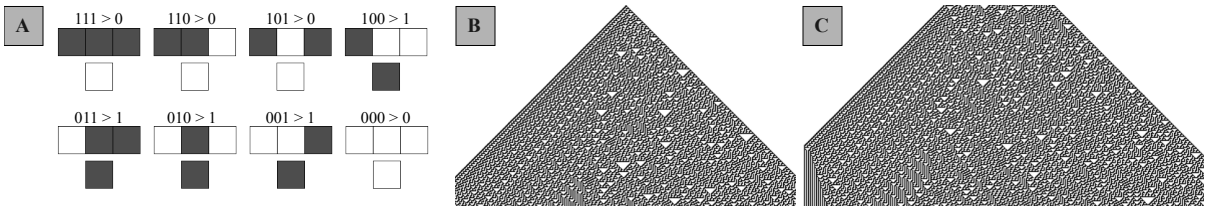


Figure 3.1: An example lookup table for Rule 30 and its evolution with different initial conditions: (A) The lookup table describing the conditions of evolution for Rule $30 = 2^4 + 2^3 + 2^2 + 2^1$. (B) 200-step evolution starting from a single black cell. (C) 200-step evolution starting from a randomized set of cells.

The elementary CA (or 1-dimensional CA) component exposes a one–dimensional lattice of binary cells that evolve in discrete time. I used two approaches for evolutions: (1) predefined rules explained by Wolfram [4] or (2) probabilistic lambda method explained by Langton [41]. A dedicated compute shader evaluates every cell in parallel, reading the three-cell neighborhood centered on each lattice site (Figure 3.1A), converting that 3-bit pattern into an 2^3-way lookup, and writing the next-state bit into a buffer. Since

the kernel is massively parallel and requires only local data, it maps extremely well to the GPU-logic. This allows for processing the entire row in a single dispatch even at 16k resolution. Toroidal boundary conditions are used so that the edge cases require no special branching and wrap naturally across GPU thread groups.

The C# side configures rules, resolution, and step-rate, then in each frame, forwards those constants into the shader. The resulting bit texture of the simulation is unpacked into a floating-point render-texture so it can be previewed directly in the HDRP renderer in Unity, captured to Spout/NDI, or fed internally as an influence map for other mixed simulations (for example, modulating pheromones in the Physarum module).

It is trivial to extend 1D CA into higher dimensions. Each cell in a 2D CA would operate under the same rule sets, except for considering a neighborhood composed of 8 cells surrounding it. The same method can be applied to 3D or higher dimensional CA without modifying the original algorithm. Just like stacking 1D rows to achieve a 2D page of elementary CA evolutions (Figure 3.1B-C), we can stack the evolutions of 2D CA to create a 3D form.

2. Cyclical Cellular Automata (CCA):

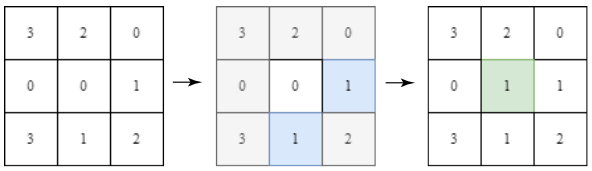


Figure 3.2: Left-to-right, a simple progression of CCA with `range`=1, `threshold`=2, `states`= 4, and *Moore* `neighborhood`.

The CCA system generalizes elementary CA into an N-state cyclic model [213]. Each pixel holds an integer state $\in [0, N-1]$ encoded in the four channels of an RGBA32F render-texture. During the update kernel, every thread examines the surrounding cells to determine its future state. Each cell counts the number of surrounding grids cells in its

`range` that hold the state equal to one more of its state (Figure 3.2). If the count meets the `threshold`, the state would be incremented by one. If the current state is already at the maximum `state`, it will cycle back to 0. While *Moore* `neighborhood` lookup considers all angles from the given cell, *Von Neumann* only considers the right-angled cells (top, bottom, right, left).

Since every frame is calculated in a single compute pass without branching, the simulation comfortably hits real-time constraints even at 8k resolutions. Parameters such as number-of-states, neighborhood radius, increment–threshold, and diffusion rate are exposed through parameter fields and can be modulated by direct control or external input such as OSC or MIDI data.

3. Flocking and Swarming Boids:

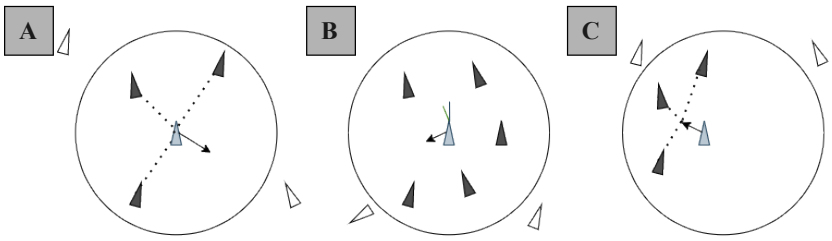


Figure 3.3: Three cases of behavior determined by surrounding Boids in a given radius: (A) Separating positions, (B) aligning directions, and (C) attracting to other's positions.

The `BoidSim` component implements a classic Reynolds flocking system [103] with several extensions. Each Boid agent holds position, velocity, acceleration and three neighbor-counters in a structured buffer. A compute kernel iterates over every pair to accumulate separation, alignment and cohesion forces, using squared ranges to avoid costly square-roots to improve performance. Food-seeking extension to original algorithm is layered on the other system logic by casting three sensors (center, left, right) into the trail map generated by Physarum or an external texture, allowing the flock to respond

to chemically evolving trails or external textures. Once forces are summed, velocity is clamped to a maximum speed and wrapped toroidally so boids exiting one edge reenter on the opposite side.

A second kernel writes colored trails into a floating-point buffer and a diffusion-decay pass blurs and attenuates them. This gives each boid a visible history and supplies the very stimulus they subsequently follow. Visualization is handled in a tiny render kernel that converts internal system values to HSV color space determined by system parameters. The entire system scales from a few dozen to tens of thousands of agents with minimal CPU overhead. However, the direct-communication logic between each boids implies a $\mathcal{O}(n^2)$ algorithmic time-complexity for the number of boids n. This results in the frame-rate dropping below real time once processing more than $\sim$40k boids on a NVIDIA RTX 3080 GPU.

4. Physarum Networks:

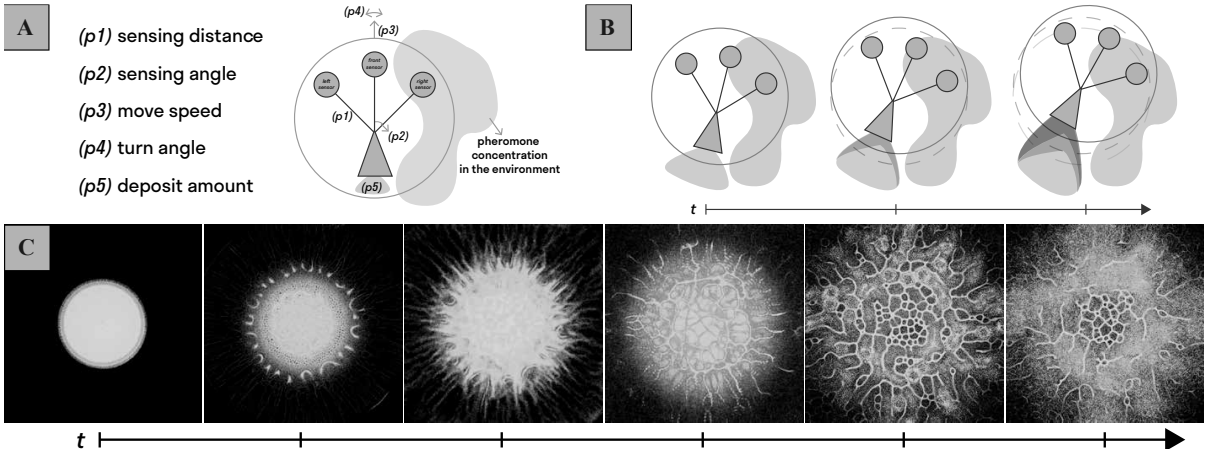


Figure 3.4: The operation of Physarum simulations: (A) Diagram showing the construction and parameters of a Physarum agent. (B) The agent moves towards higher pheromone concentration based on parameters $p1...p5$. (C) An example animation unfolding over time with 1 million agents that leave bright pheromones.

The `PhysarumSim` component models slime-mold growth via millions of micro-agents

that deposit and sense pheromones [104]. Each compute thread updates a single agent (Figure 3.4A): the direction vector is rotated by a sensor-derived angle (left, center, right sampling), moved forward (Figure 3.4B), and wrapped on a toroidal surface. Figure 3.4C shows a sample animation sequence of ∼1 million agents unfolding over time. These agents start from a circular initial position with randomized directions, and they move by depositing white pheromones in the environment. Four independent species each with its own sense-distance, turn-angle and movement speed arrays can be packed into one render-texture distributed over RGBA channels. This enable a complex inter-species interaction, variable visual behavior, and distinct colored pheromone fields. An optional external-influence texture that might be supplied live via Spout, NDI or any other simulation can influence the pheromone map, therefore affecting the movement behavior.

Trail deposition, global diffusion, and evaporation are executed in their own compute kernels so the ratios can be tuned at run-time. A high-resolution ARGBFloat buffer preserves fine gradient detail, and a final render kernel converts agent trails into color using per-species HSV parameters. The design avoids discrete color operations by letting overlapping deposits accumulate naturally in float space, and compensates with a lightweight clamp at the time of render. Since this simulation works with stigmergy communication rather than agent-to-agent direct-communication like the boids, its algorithmic time-complexity is linear $\mathcal{O}(n)$ and results in comfortably handling ∼40 million agents on an NVIDIA RTX 3080 GPU. This still leaves enough bandwidth for other concurrent calculations such us flocking or CA simulations.

Mixed-simulation Environments

Here, I describe strategies for combining core algorithms with one another to produce novel and more refined visual results that were not feasible using a single system.

Custom environments containing a mixture of these systems modulating each other's

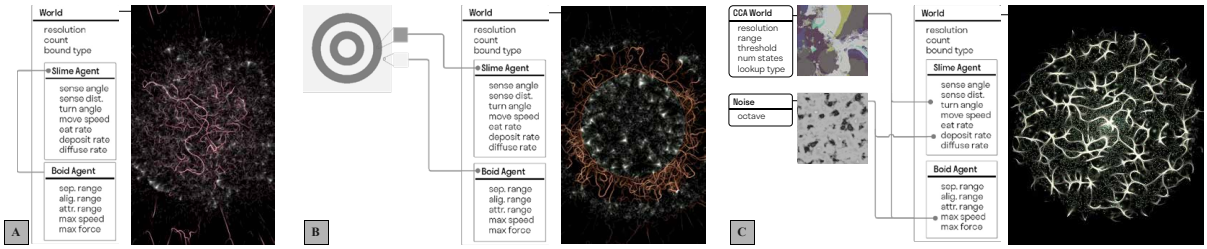


Figure 3.5: Three approaches to construct a mixed simulation approach to composing different algorithms. (A) Boid agents interact with Physarum agents; (B) a static map modulates an agent's movement logic from Boid to Physarum; (C) dynamics maps constructed from CCA and Perlin noise modulates the parameters of Boid and Physarum agents.

parameters contributed an additional behavioral complexity to the visual outcomes. I give three examples of this approach: In Figure 3.5A, Boid and Physarum agents live in the same environment, but only Boid agents respond to individual Physarum agents. Even this simple addition provides a more diverse aesthetic. Figure 3.5B shows examples of spatially modulating the agent type from Physarum to Boid based on an invisible *static* map, which can contain vector or raster graphics. The bright pixels correspond to parts where the agent behaves like a Boid agent, and the gray corresponds to Physarum agent behavior. Figure 3.5C shows an example setup using *dynamic* maps to spatially modulate the simulation parameters. An invisible CCA simulation with its own set of parameters and an animated Perlin noise is routed into the mixed-simulation manager where they can seed Physarum growth or act as a dynamic food-map for boids, creating visually rich feedback loops between discrete-state CA and continuous agent fields.

After experimenting with custom methods for combining system outputs, I formalized this method by implementing a manager class capable of combining simulations in different configurations. The `SimulationManager` class orchestrates all sub-systems and composites their outputs into a single texture for displaying or streaming. During reset it allocates a shared resolution for every child simulation, initializes their buffers, and then spawns a composite-render kernel that adds the Physarum and Boid channels with pre-

served alpha. Each Update tick can execute multiple simulation steps (`stepsPerFrame`) under a modulo gate (`stepMod`). This allows time warping the simulation to slow-motion or time-lapse playback with synced clocks between different components. In addition, the manager acts as a routing hub: Frames collected through `SpoutReceiver` or `NDIReceiver` can drive `externalInfluenceTexture` on simulations, MIDI or OSC triggers can trigger simulation parameters, and the resulting composite can be sent out via Spout or NDI for external modifications or projection mapping. Because every simulation shares the same coordinate space and texture formats, adding more layers to this manager is straightforward by instantiating components and textures, and connecting them in the manager's inspector.

3.2.2 Strategies for Parameter Control

System design and implementation is only one part of the creating works with generative systems. Expressive generative systems contain a vast space of possible outcomes; each set of initial conditions, system parameters, and modulating these parameters in runtime through interaction produces dramatically different results. However, not all regions of this possible output space lead to visually compelling results. For example, most executions of CCA algorithm did not produce spirals and waves. *Discovering* aesthetically appealing results from this set is a creative practice by itself.

This subsection reflects on how I distributed my creative agency within these systems by developing both computational and manual methods for parameter space exploration. These strategies emerged through technical experimentation as well as conversations with other practitioners. The discussions in the online community revealed insights regarding algorithmic control schemes and manual parameter tracking strategies. Some technical artists from the community suggested using Langton's λ-parameter into CA algorithms.

Some artists described methods for screen recording the parameter window and the visual output at the same time. In addition, my discussions with the expert composer and image-maker revealed that parameter space exploration is similar to playing a musical instrument, requiring mastery over the system configurations to find desirable visual compositions.

Computational Method: Phase Transitions at the Edge of Criticality

Beyond manual tweaking, computational techniques for parameter space analysis provided a way to identify promising regions. In particular, Langton's research on CA behaviors [41] not only inspired this project's name but also motivated me to algorithmically explore global parameters that affect future iteration's 'survival rates'. This approach helped me finding desirable results quicker, allowing for deeper exploration with less frustration.

I adapted Langton's λ-parameter concept to systematically investigate the parameter spaces of CA. This parameter describes the likelihood of variety in the rule definitions; $\lambda = 0$ means all rules result in a *dead* cell, and $\lambda = 1$ means all rules result in a *alive* cell. Parameter ranges around 0.5 to 0.7 produced the most intricate and dynamic patterns.

Manual Methods: Real-time Exploration and Recording

Most of the visual results presented in this chapter emerged through active exploration of these systems and parameter spaces. I treated them as real-time instruments, improvised with their controls, and recorded parameter modifications and outputs in real-time.

Unity offers a way of creating custom user interfaces for the system components through `SerializeField` and `Range` functionalities. I identified useful parameters that affect the system evolution and created serialized fields to be able to modify these values in

system run-time. In addition, I also mapped the same parameter values to MIDI controls to engage with instrument approach in a more tangible way. When using the on-screen sliders, I recorded the UI and the visual output at the same time to review the effects of each parameter on the emergent behavior. In addition to discovering the appealing parameter ranges manually, I also created various *presets* containing a library of visual behaviors. This allowed me to revisit certain configurations without going through the discovery phase again.

3.3 Case Study: *Simulacra Naturae*

This prototype is a 9 m x 6 m data-driven media installation that explores artificial life in a situated format. It makes use of the *Edge of Chaos* system as a layer to other life-like algorithms, and drives them using pre-recorded data collected from human brain organoids. These biosignals modulate emergent agent behaviors of termite colonies and slime molds. The spatial environment, filled with live tropical plants, invites participants into a sensory field shaped by nonhuman presence. In this chapter, I address the *digital visual components* of *Simulacra Naturae*; its physical and cyber-physical aspects are elaborated in Section 6.2.1-Future Directions to highlight connections to distributed agencies.

The project's title was inspired by Lucretius' notion of *simulacra*, atomic films that radiate from all things and mediate human perception [214]. Building on Lucretius and Latin word *simulacrum* that stands for *likeness*, *Simulacra Naturae* builds a 'likeness of nature' that mimics processes found in nature. This is further operationalized by the use of biosignals and real plants in the installation.

Simulacra Naturae is driven by a dataset of neuronal activity recorded from human brain organoids at the Kosik Neurobiology Lab [215]. I was not involved in data collection

or the collaboration with the neuroscience team. These organoids were cultivated from stem cells and interfaced with electrodes to capture activity, and the resulting raw data were sorted to yield 131 active channel neurons (for detailed procedures, refer to [48] and [210]). 131-channel active neurons serve as the primary dataset for this system. This neuronal population has a subset of 27 units, referred to as backbone neurons, that demonstrated consistent temporal coordination and strong interactivity. The full set of 131 neurons informs the real-time behavior of the digital simulations.

3.3.1 Visual System Design and Integration

The primary visual component of *Simulacra Naturae* is driven by a virtual ecosystem of various simulated life-like agents. These simulations include the Physarum and Boid agents from the *Edge of Chaos* interacting with termite agents. This digital ecosystem is projected on a large scale wall with 4K laser projectors. Figure 3.6 shows two still frame exports of the video feed and three close-ups taken at different times.

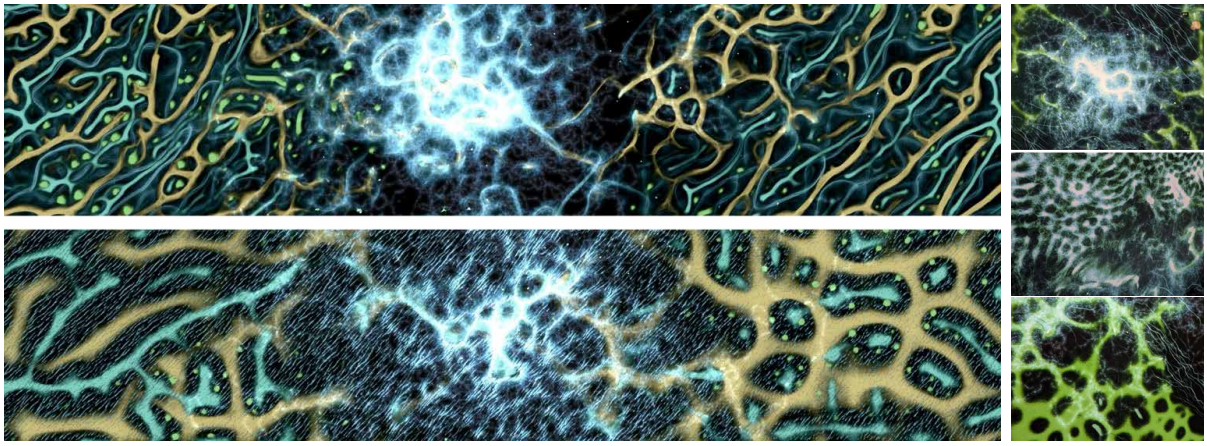


Figure 3.6: (Left) Two exports of projected visuals combining the termites, slime molds, and flocking agents: The bright regions in the middle represent termite mounds. Orange, green turquoise and light blue networks represent Physarum pheromone trails, and small local activity around the mound represent boid agents. (Right) Closer photos of the variations in the projected visuals running with various parameters.

The agent-based termite simulation, implemented by my collaborator Nefeli in Pro-

cessing, follows self-organization and decentralized coordination principles. Each agent, corresponding to a neuron in the organoid dataset, follows local behavioral rules, including stochastic motion, environmental sensing, and conditional trail deposition. Agents interact with a scalar field environment that retains memory through dynamic marking and decay, similar to pheromone trails. This feedback-rich loop generates emergent behaviors like clustering, trail formation, and spatial patterning without centralized control. Agents operate autonomously, but real-time firing events from the organoid dataset intervene, depositing trails at spikes and overriding endogenous behavior. Trail marking becomes a shared parameter activated by local environmental feedback or exogenous signals. The system integrates spontaneous neural activity into a self-organizing simulation, modulating agent behavior and collective spatial dynamics.

Building upon this model of decentralized behavior, the slime mold (*Physarum polycephalum*) simulation introduces another self-organizational layer in the digital component. The system implementation is explained in detail in Section 3.2.1. The slime mold simulations in *Simulacra Naturae* form a synthetic ecosystem where virtual physarum agents deposit chemical signals (pheromones) and respond to existing signals in the environment in two ways: (1) Modulating the primary parameters with pre-recorded brain organoid signals, and (2) responding to termite simulations discussed above.

The other a-life layer is the flocking systems, evoking the collective movement of birds or fish. The system implementation is explained in detail in Section 3.2.1. In *Simulacra Naturae*, digital agents adjust their velocity and direction in real-time based on signals derived from live neuronal populations, effectively becoming extensions of the organoid's temporal rhythms. This results in fluid visual behaviors that echo biological swarm intelligence.

These systems have been integrated using TouchDesigner [216]. The patch parses neuronal recording and distributes it over OSC for other machines. *Edge of Chaos*'s

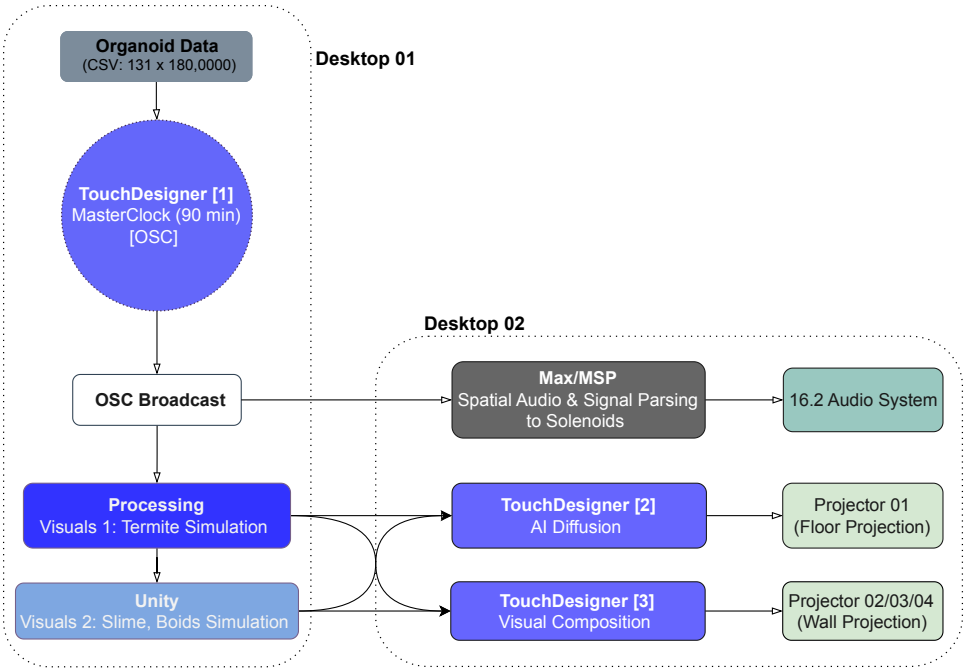


Figure 3.7: Interaction diagram of the system. Organoid data is parsed in the first TouchDesigner patch that transmits the data through OSC and MQTT. The OSC data is captured by Unity, Processing, second TouchDesigner patch running Generative AI visuals, and the Max/MSP patch governing the soundscape.

outputs from Unity and termite simulations from Processing sent out using Spout, and combined in TouchDesigner patch. Projection alignment and warping were performed using TouchDesigner's native tools. Figure 3.7 shows a signal flow diagram containing all components in *Simulacra Naturae*. These components are outlined in Section 6.2.1.

3.3.2 Data-driven and Responsive Composition

In this project, I extended the *Edge of Chaos* system to support external data streams and to compose simulations running across different applications and machines. This enabled a responsive visual system that adapts to incoming signals in real time and produces dynamic behavior with varying parameters.

The visual engine accepts external render textures of arbitrary size via Spout (Windows) or NDI (cross-platform) data streaming technologies. Incoming frames are cap-

tured, converted to a GPU texture, and routed as influence maps to system parameters. These textures modulate exposed fields in *Edge of Chaos* components (e.g. pheromone deposition, boid steering weights, CCA thresholds) and can be sampled directly in compute shaders. I assigned a selection of physarum agents to repel, others to attract the termite mounds. When resolutions differ, inputs are scaled on-GPU to the simulation space to avoid CPU bottlenecks. This pathway allows any upstream process that can output a texture to drive the system.

I generalized the mixed-simulation approach described in Section 3.2.1 to include simulations running in other software on the same or different computers. Termite agents authored in Processing, Unity-based *Edge of Chaos* modules, and auxiliary processes in TouchDesigner exchange data as textures over Spout/NDI and as lightweight messages over OSC/MQTT. Treating external simulations as system inputs increases the diversity of interacting processes and makes collaboration with other practitioners straightforward. This distributed setup widens the space of computational agencies that can co-shape the final image.

Pre-recorded brain organoid activity modulates system parameters in real-time, producing a composition that evolves with the data. Population firing rate and event markers are mapped to higher-level controls (e.g. global diffusion/decay, species turn angles, trail brightness) and to transitional triggers. This mapping produces visible shifts in pattern density, localized structures, and synchronized motion. The result is a controllable and responsive composition pipeline that remains dynamic and sensitive to ever-changing input signals.

3.4 Results

Here, I present the visual results of *Edge of Chaos* and *Simulacra Naturae* and discuss how they help evaluate distinct modes of engagement with the system. This artifact-driven evaluation highlights how parameter exploration, algorithmic composition, and spatial rendering contributes to the emergence of intricate, self-organizing patterns and complex structures. Together, they demonstrate how system design enables artists to shape generative behaviors into expressive artifacts.

3.4.1 Parameter Space Characterization

I use Physarum simulation as an example to talk about how I characterize parameter spaces. Over time, I noticed I developed preferences towards using certain parameters over others. While I change the main behavioral parameters often, I rarely change, for example, the pheromone diffusion rate. This is due to how disruptive these parameters can be in the final rendered image. Even 1% modification to diffusion rate can make the image over exposed when there are thirty million agents in the scene. However, the behavioral parameters provide the most useful parameterization for finding emergent behaviors of in Physarum simulations. These are `moveSpeed`, `turnAngle`, `senseAngle` and `senseDistance`. Figure 3.8 compares evenly sampled values of sense distance and angle with respect to each other. As sense distance increases, the pheromone trails tend to accumulate at higher distances, producing bigger-scale networks. Sense angle, on the other hand, shows a phase transition as the value goes higher than π degrees, switching from an orderly behavior to a more chaotic region.

The results seen in the first column and row Figure 3.8 represent special values of Physarum parameters. I noticed when these values are small, or `moveSpeed` is high, the agents cannot settle on a meaningful pheromone trail and disperses through the canvas.

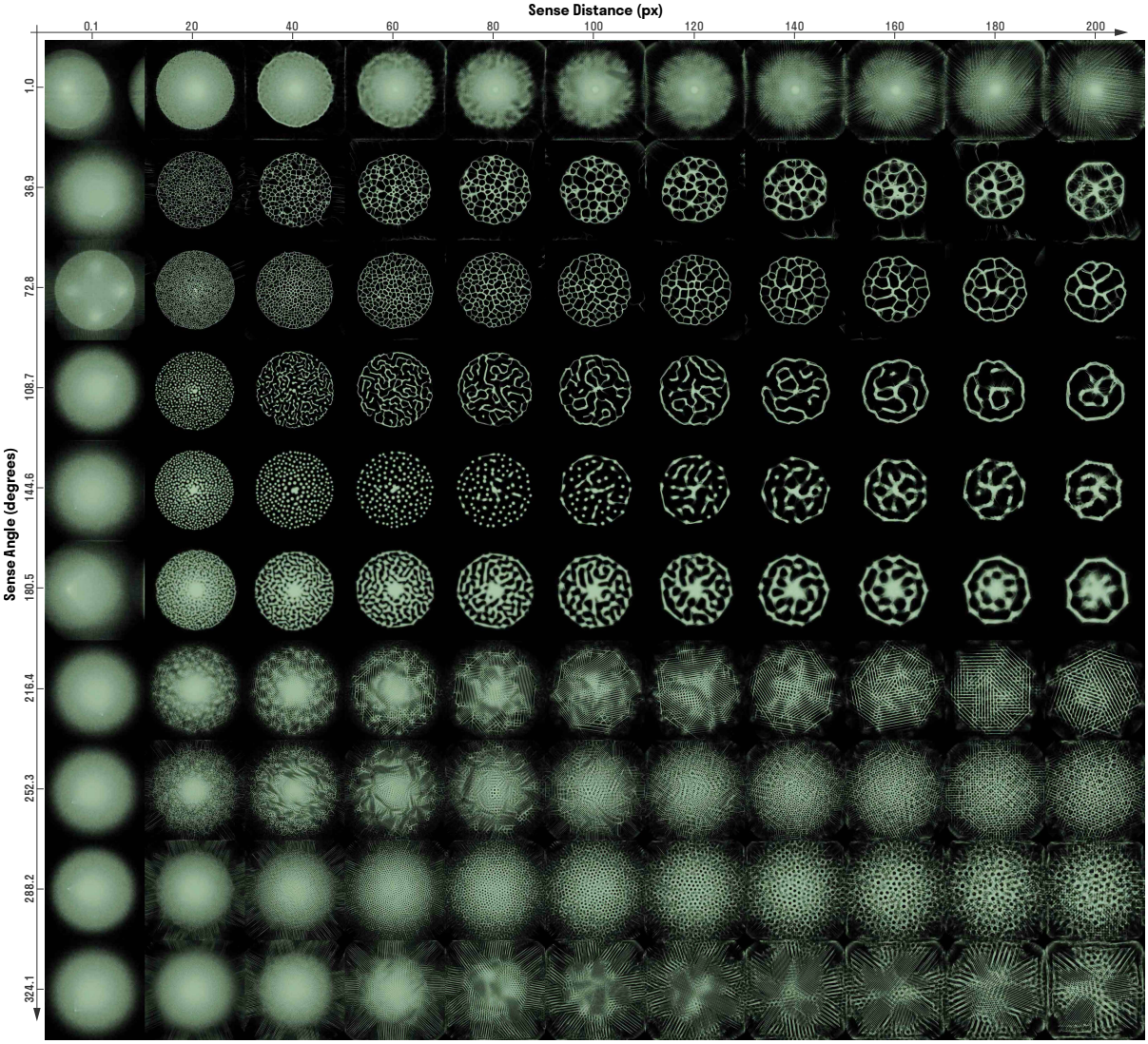


Figure 3.8: Example patterns emerging in Physarum simulation with uniform sampling of two parameter ranges (X-axis: `senseDistance` $\in [0, 200]$ px, Y-axis: `senseAngle` $\in [1, 360]$ degrees). `senseAngle` displays a phase transition at around π degrees. Each 4096x4096 frame contains 20 million agents starting in a circular distribution and is captured 3-seconds after reset, with fixed `moveSpeed`=5 and `turnAngle`=45.

I used this property to advantage and created a background layer with these parameter ranges. Figure 3.10B shows this use case in action, where the bright backgrounds are composed of fast moving Physarum agents.

3.4.2 Emergent Outcomes by Algorithm

All four algorithms explained in Section 3.2.1 are capable of producing high-fidelity, two dimensional outcomes. I primarily worked in 2D formats and they are most abundant in the artifacts I produced. However, algorithms except for Physarum simulations can run in 3D worlds using voxels instead of pixels. I kept the Physarum to 2D because its computational agency arises from the high-fidelity pheromone trails, and voxel representation is not conducive for this resolution.

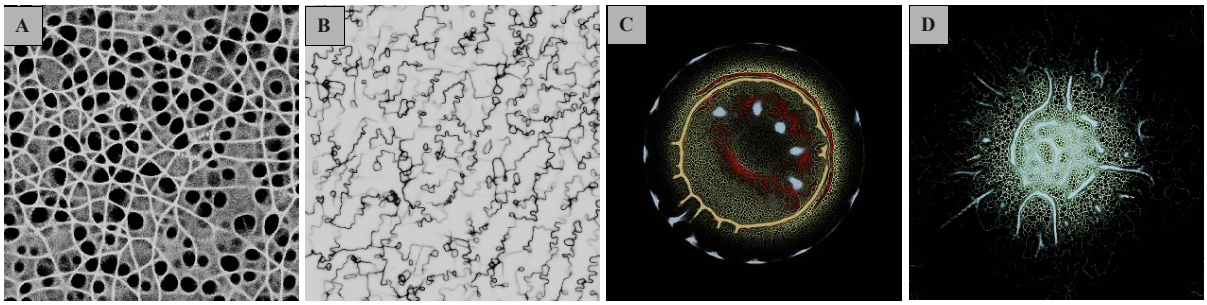


Figure 3.9: Visual examples from Physarum simulations: (A-B) Results of simulating 1 million agents that leave bright and dark pheromones, respectively, without spatial constraints. (C-D) Results of simulating 50 and 30 million, respectively, agents of four types, differentiated by distinct-colored pheromones, with a circular spatial constraint.

2D Behaviors: I discovered the most expressive algorithm I used in *Edge of Chaos* was the Physarum simulations. Their granular pheromone-based behavior allowed me to explore various collaborative approaches, and I chose not to convert them to 3D. The primary emergent form of Physarum algorithm is the maze or worm-like strands that forms through pheromones left behind as the agents move in the virtual environment. For example, Figures 3.9A and B show these structures emerging with one million agents

and mid-range valued parameters. As I increased the number of agents and assigned them different types denoted with various colors, the visual compositions became much more complex. Figures 3.9C and D show four types of 50 and 30 million agents, respectively, initiated from a circular starting region in the middle.

When I added the ability to consume other agent type's pheromones, a power relationship emerged. `depositAmount` and `eatAmount` parameters determined which type was dominant in the composition. As I used this system as an instrument, I started discovering the ranges of parameters that yield interesting behavior. Figure 3.10 shows three examples of this type of behavior that produced biological-looking forms. In multi-agent

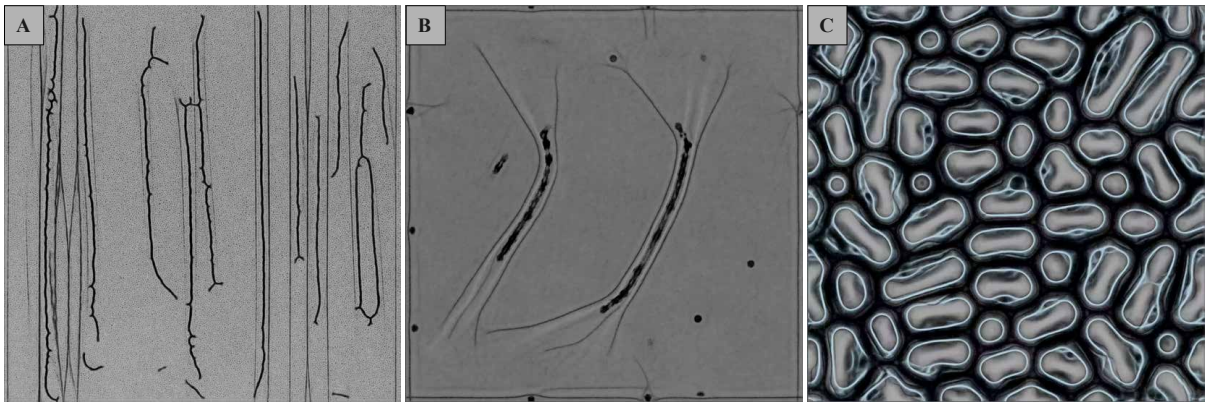


Figure 3.10: Cell-like examples from Physarum simulations containing multiple agent types. Keeping one agent's parameters at extreme values makes it disperse through the scene and acts as a background. (A) Bright-colored agent's `senseDistance` is high and `turnAngle` is low. (B) Bright-colored agent's `moveSpeed` is very high, and dark-colored agent's `eatAmount` is high. (C) Black regions formed by an agent whose `eatAmount` is very high, constraining other three agents in bubbles.

systems, I found taking one agent's parameters to extreme ranges often resulted in its widespread presence as a vibrant backdrop. I observed this behavior with `senseDistance` and `senseAngle` (as explained above), as well as high `moveSpeed` values. Increasing one agent's `eatAmount` and others `depositAmount` also resulted in a feedback mechanism where they fed off of each other and produced bubble-like structures shown in Figure 3.10C. These simulations provide an intriguing insight into the complex behaviors

and interactions of these simulated entities.

CA yielded less expressive results compared to Physarum simulation, but the control strategy using the λ parameter was inspiring. In Figure 3.11A, I show a particular run of this algorithm starting from a randomized line at the top of the canvas and evolves with a particular seed that determines evolutionary rules based on the λ parameter.

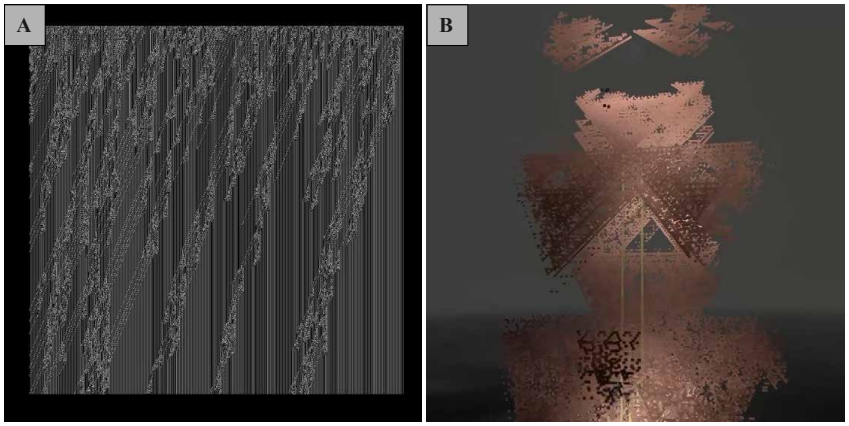


Figure 3.11: Visual examples of elementary CA algorithm: (A) The algorithm evolves with a random starting condition and a mid-range λ value (around 0.6), demonstrating longer transients without collapsing into complete order or chaos. (B) A 2D CA evolution is visualized as stacked voxels after removing most abundant states to reveal an emergent structure (i.e. Sierpinski triangle) in time axis.

I adapted a variant of CA to use cyclical evolution logic. Similar to CA, CCA operates in a grid-based lattice, however, as shown in Figure 3.12, it produces spiraling geometry resembling the Belousov-Zhabotinsky (BZ) reaction in chemical morphogenesis due to its update conditions (see Section 3.2.1). When I ran this algorithm as described in [213], I observed interesting patterns, however the spiraling patterns grew out without constraints and canceled each other. In this setting, most of the parameter configurations quickly dissolved into complete order. An artist in Arsiliath's online community posted a method of 'freezing' one state of the CCA algorithm and producing maze-like geometry in the environment. Using this as a map for a second CCA simulation produced vein-like patterns. The same spirals from my initial attempt were running through channels

created by the frozen environment. Figure 3.12A shows a still of this strategy (i.e. the darker 'walls' between the bright regions are the frozen environment elements).

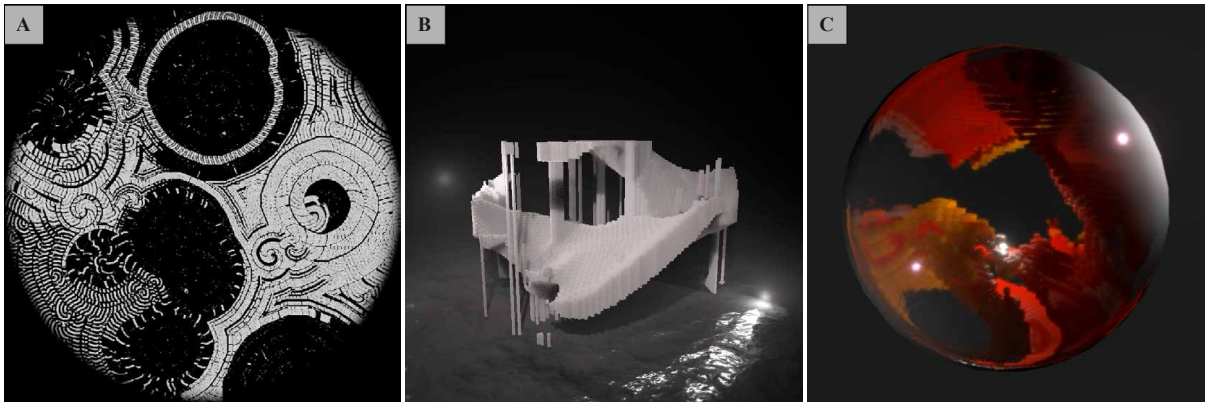


Figure 3.12: Visual examples of cyclical CCA algorithm: (A) Vein-like structures emerging from two-staged execution of the algorithm. (B) Stacking the evolution of a 2D CCA in Z-axis visualized by selectively disabling most abundant states. (C) 3D CCA visualized inside a manually-added refractive material; only two states are visualized for visual clarity.

And lastly, the behavior of Boid agents in flocking algorithm is dictated by three parameters: `separationRange`, `alignmentRange`, and `cohesionRange`. These parameters can produce vastly different results, and I experimented with various visualization techniques to render emergent patterns in a custom fashion. As shown in Figure 3.13B, using a dynamic color palette determined by the number of agents falling within the three parameter ranges was fruitful. In addition, I experimented with scene-level techniques, Figure 3.13A shows an example where I overlaid four planes in Unity and rendered the alpha-enabled simulation output to all four geometry simultaneously. In addition, I also added food seeking behavior to flocking agents, governed by `eatAmount` and `depositAmount` parameters. This addition made Boid agent behavior more akin to Physarum agents.

3D Forms: In *Edge of Chaos*, I experimented with creating 3D geometry using the algorithms described in this chapter. The benefit of using Unity was the additional render

utilities, such as subsurface scattering, external 3D mesh support, ray traced shadows and reflections, that make the visuals high-fidelity. I use three strategies to construct three-dimensional geometry:

Figures 3.11B and 3.12B calculate 2D CA and CCA, respectively, and stack them in Z-axis to construct three-dimensional forms. This approach embeds the temporal axis into one of the spatial axes. It is also feasible to simulate a 3D representation by

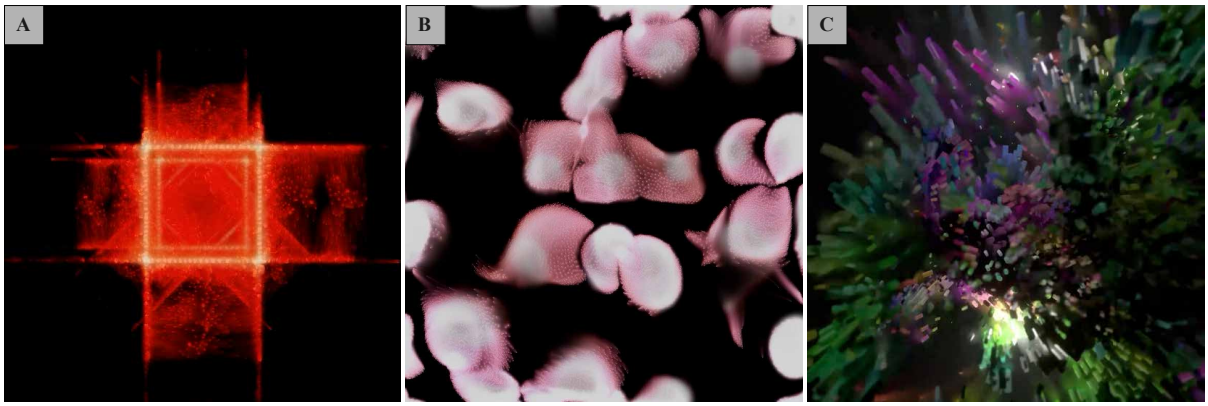


Figure 3.13: Visual examples of flocking algorithm: (A) Symmetrical overlapping placement of four planar geometries with the same material, using the 2D Boids algorithm output with an alpha channel. (B) 2D flocking behavior displaying emergent petal-like patterns, dynamically colored based on the number of agents falling within the parameter ranges. (C) Running the same algorithm with 3D vectors in an immersive environment; trails are rendered by customizing the virtual camera to keep the previous render in the viewport.

incrementing the spatial dimensions. In this case, visualization becomes tricky. In my examples, for example in Figure 3.12C, I remove certain states of the algorithm from the visualization to create gaps in 3D structure, use a voxel field to create the geometry, and create a glass-like surface in Unity to encapsulate the created form in a sphere. Lastly, two examples on the right show Boid agents running in a 3D space, as shown in Figure 3.13C.

In practice, each strategy offers different affordances. Stacking 2D slices along Z embeds time into space, revealing long-transient motifs and repeated structures that are

hard to perceive in a single or animated sequences of 2D slices, as demonstrated by the emergence of Sierpinski triangles in Figure 3.11B. For the 3D CCA, rendering all states produces an opaque mass, but selectively disabling the most frequent states exposes cavities. For 3D Boids, I accumulated trails by keeping the previous frame in the camera buffer, which makes flocks global behavior more legible at a scene scale.

Using voxel representations for 3D also introduced constraints on the performance. Voxel fields beyond 512^3 quickly became memory-bound on a single GPU, so I mainly used mid-resolution grids with selective state culling. As mentioned, I did not port Physarum to 3D because its expressive agency depends on high-fidelity trail networks, and at these practical voxel resolutions, pheromone details are lost. For 3D scenes, I used voxel resolutions $128 - 256^3$ and 5k Boids with trail accumulation on an RTX 3080, with Unity HDRP handling lighting and refraction.

3.4.3 Presentation Context

The *Edge of Chaos* outcomes has been exhibited as framed prints accompanied by vertical 4K screens as part of *SyMADES* exhibition in Santa Barbara, CA in 2022 [45]. The purpose of using still frames to present a real-time system was to emphasize their visual compositional qualities. Displaying the emergent patterns as still images allowed audience members to contemplate on self-organization and complexity. I placed two vertical screens in the center of still frames, showing animated recordings of my workflow generating the emergent compositions (Figure 3.14-left). These videos contained the temporal dimension missing in the still images, and demonstrated the evolutionary aspect of the co-creative process with computational systems. I had a chance to exhibit these prints in a larger wall as part of *Deep Cuts* exhibition in 2025 [46]. This time, I used a similar setup containing still images and animated recordings, with the exception that



Figure 3.14: The installation of *Edge of Chaos* with still frames and video screens in Elings Hall, UCSB (left) at *SyMADES* in 2022 and (right) at *Deep Cuts* in 2025. The smaller frames on the right showed evolutionary nature of emergent algorithms that required closer investigation.

I printed new frames to accompany previously produced frames. I placed the animated recording facing away from the wall with the prints because I wanted to convey the narrative of temporal evolution and continuity only using the still images (Figure 3.14-right). To achieve this, I selected evolutionary frames from my high-resolution recordings, and assembled them on the wall side-by-side with the previous frames. This approach meant to demonstrate the temporal nature of the algorithms in spatial dimension.

We exhibited *Simulacra Naturae* as a data-driven, large-scale piece as part of *Deep Cuts* exhibition in Santa Barbara, CA in 2025 [47]. Figure 3.15 shows two versions of the work in different environments: Left image shows indoor projections in TransLab's 9 m x 6 m environment with tropical plants, and right shows projection mapping of the visuals onto two three-story buildings. The visuals initially developed in *Edge of Chaos* project has been used at this resolution (10,184 × 2,160 pixels) and dimension for the first time. It highlighted a few important compositional principles. The GPU-driven architecture allowed for seamless integration at this resolution while supporting more than 30 million agents running in real-time. The increased dimension required localized emergent visuals and diversity. To achieve this, I inverted the toroidal space so that, for instance, bottom-left of screen would wrap to top-right instead of the top-left portion of



Figure 3.15: The installation of *Simulacra Naturae* at *Deep Cuts* exhibition (left) in TransLab and (right) at SBCAST.

the display surface. The other modification was the integration of termite simulations. We positioned the termite mounds in the middle portion of the rectangular display so that certain Physarum and Boid agents would repel from it and create variation in the global patterns.

3.5 Discussion

This discussion reflects on the strengths and challenges of *Edge of Chaos* as a generative compositional system, situating its technical affordances and limitations within broader questions of computational creativity and artistic practice.

3.5.1 GPU-driven Framework Permits Increased Emergence

The GPU computation permits rules to unfold over an extended temporal space, therefore increasing the probability that rare, high-complexity spatial forms will emerge. From a theoretical standpoint, the observations of GPU compute align with complexity

science's fundamental principle where application of simple rules many times can produce more chances for system components to interact and self-organize [6]. In other words, finer spatial–temporal granularity provides more micro-states from which macroscopic order can emerge [41, 4].

In sequential computing on central-processing units (CPUs), each line of code runs on one core and waits for the previous line to finish executing. Processing 2D/3D media sequentially introduces significant overhead and often results in *lower resolutions*, *slower execution*, and *interactivity challenges*. Modern GPU architectures can overcome these challenges and have transformed how researchers and artists explore self-organizing complex systems. Their Single-Instruction-Multiple-Data (SIMD) architecture offers massively parallel arithmetic units. Each core operates at a higher rate than traditional CPU cores due to specialized arithmetic units. These features allow millions of microscopic system interactions to be evaluated simultaneously.

This computational architecture addresses the challenges of sequential computing for emergent-behavior research in three key ways:

Scale: All algorithms of *Edge of Chaos* involve highly *parallelizable media* governed by *local rules*. 1D structured buffers (agent states), 2D render textures (environment states or final renders), and 3D voxel fields are all suitable for passing into a compute shader kernel that can dispatch and index into up to three-dimensional structured GPU cores. Since the algorithms execute rules locally, I can use the structured identifiers of dispatched cores to index into GPU memory and retrieve neighbor states to calculate local interactions instantly. These features allow increased scale in both output resolution and agent count. Higher resolutions reveal fine-grained visual phenomena, such as spirals, waves, and filaments, that remain invisible at coarser resolutions. Larger agent populations directly increase system's component interactions and raises the likelihood

of emergent complexity (Figure 3.9).

Speed: The capacity to rapidly iterate thousands of discrete time steps aids in discovering stable emergent forms in systems requiring longer executions. Producing over 200 frames per second, combined with the ability to execute multiple simulation steps per frame using the `stepsPerFrame` variable, enabled faster production of intricate visual results. This speed directly supports real-time parameter space exploration.

Real-timeliness: *Edge of Chaos* leverages the GPU's parallel architecture to increase visual complexity without compromising real-time *rendering* and *interaction*. Keeping calculations within GPU memory reduces data transfer bottlenecks, while Unity's high-definition render pipeline (HDRP) supports advanced effects such as sub-surface scattering and real-time shadows. This real-time simulation and rendering facilitate immediacy and enable manual parameter exploration to become a performative, instrument-like engagement. This was crucial in *Simulacra Naturae*'s data-driven and responsive setup. Responding to other system's outputs and organoid firings synchronously enabled an immersive environment.

I argue that GPU computing is conducive to increased chance for emergence for visual media. Most interesting results were revealed either at high resolutions, or when simulation has evolved extensively. This framework enabled me to explore, control, and communicate complex and self-organizing phenomena found in artificial life algorithms in a high-resolution and real-time manner.

3.5.2 Parameter Exploration as a Creative Practice

The behavior of creative systems typically has a large number of executable pathways [39] and a vast amount of parameter combinations that shape the execution. Yet, portions of this parameter space result in more aesthetically appealing outcomes than others.

Exhaustive traversal of the parameter space presents both pragmatic and technical challenges for the artistic use of computational systems. Assessing the aesthetic qualities of every possible artifact would require allowing each parameter combination in the dynamical system to evolve fully to its terminal state. For many systems, this exhaustive search is infeasible on contemporary hardware. Moreover, a human artist cannot feasibly evaluate the effectively infinite number of possibilities. Therefore, algorithmic methods that *aid* human artist to reduce parameter spaces becomes a viable approach.

I consider parameter space exploration of a generative algorithm as an artistic act that requires time and labor investment and enables a collaboration with computational processes. This computational creative practice involves four stages: (1) developing the system's processes, (2) visually analyzing its execution pathways, (3) clustering parameter spaces that produce similar results, and (4) defining higher-level control parameters to structure subsequent exploration. Once these parameters are defined, the artist can improvise with higher-level controls for overarching system behavior and lower-level parameters for fine-grained modulation. Using this manual-computational control approach, this pipeline would effectively *instrumentalize* the algorithmic processes and distribute the creative agency between the computational system and the artist. In this way, the system itself becomes an instrument, invites the artist to master its affordances and constraints through an embodied and performative practice.

3.5.3 Limitations

While *Edge of Chaos* provided a rich playground for exploring generative systems and creative agency, several limitations surfaced during its development and artistic use.

Over-reliance on established system aesthetics in generative algorithms risks reducing visual originality. To address this, I implemented custom features in existing algorithms,

for instance adding a food-seeking behavior to Boid agents, and used the system as an instrument to identify original aesthetic behaviors, such as increasing Physarum agent movement speed to create a dynamic background for other agents.

The subjectivity of what constitutes 'interesting' or 'original' results remains an open question in evaluating artifacts produced by these systems. Although the results were critiqued by expert practitioners and exhibited publicly, aesthetic judgments are inevitably grounded in personal and cultural frameworks. Building on my own art practice while drawing from insights of collaborators helped broaden these perspectives, but the evaluation of generative work remains highly context-dependent.

There are also compute limits inherent to real-time simulations. For example, the flocking algorithm begins to run below real-time performance with approximately 20,000 agents due to its $\mathcal{O}(n^2)$ time complexity. Similarly, Physarum simulations start running below 60 frames per second with more than ∼50 million agents on an NVIDIA RTX 3080 GPU (Figure 3.9C). While these limits did not constrain the variety and originality of my visual results in most cases, future work could explore advanced spatial partitioning techniques to improve scalability in dense simulations, particularly for flocking algorithms.

To summarize, this chapter demonstrated how computational complex systems can be designed and controlled to create emergent visual artworks. This was discussed first through the *Edge of Chaos* and then in a data-driven, large-scale deployment with *Simulacra Naturae*. While these studies focus on digital systems, tangible media provide different affordances for perception, interaction, and aesthetic outcomes. Holding a tangible object allows for a more nuanced understanding of its form, enabling comprehension of its abstract qualities and formal properties. The subsequent chapter examines how

digital form generation interfaces with physical production to explore the creative agency of materials.

Chapter 4

Distributing Agency between Human and Material Actors

Another component of emergent expression is the *material agency* present in the physical production. Material agency differs from computational agency in how it is integrated into the creative process. In Chapter 3, I described strategies of co-designing with computational agents and intention to offload some of artist's design intent onto emergent processes. However, when an artist works with physical materials, they collaborate with physical materials whose material agency is reflected in the artistic workflow as an unintentional quality. This implies that an artist engaging with physical production requires to design with this type of creative agency in mind. Including physical materials and machines to execute digital instructions in physical form introduce an exponential increase in the number of actors influencing the final outcome.

When material agency is integrated into a computational art practice, it becomes critical to understand *(1)* the manual practice involving the chosen material and *(2)* how software development changes in the face of the unique conditions of the material practice. This chapter discusses how *ceramic craft practices* can feed into new computational

workflows and methodologies for creating tangible objects through the use of digital fabrication technologies. The material agency of natural materials like clay complicates and enriches the distribution of creative agency. To better isolate the interactions between the material and human agencies, this chapter does not rely on generative digital algorithms but focuses on manual/parametric digital design methods.

This chapter primarily addresses RQ3 by identifying clay pipeline steps that are suitable for supporting with manual-computational methods. It details a long collaboration between ceramic artists and researchers that revealed software development principles for machine interactions, form production, and surface texturing. It also details the development of an adaptable manual-computational workflow for surface ornamentation that demonstrates how rapid digital fabrication and non-linear material properties can be integrated. Through the development of these tools and workflow, it also contributes to RQ1 by identifying areas in which unpredictable qualities of clay yields to morphogenetic making. It also partially address the RQ2's distribution of creative agency between material and human actors by identifying how materials dictate and shape the design of an artifact, and how professional and amateur designers work with this type of agency for their final works.

4.1 Methodology

In this chapter, I use HCI research methodologies for developing creativity support technologies [174] to gain insights into material production workflows and to support the creative agency of expert practitioners. Developing software tools for artistic production presents an inherent tension between building software and making art. This is especially true in the context of digital fabrication, where tool development must balance implementation with the production of physical objects for testing due to the difficulty



Figure 4.1: Prior works of participating residents: (A) Hand-built vessels and 3d-printed flasks with wild-fire ash glaze by Avi Farber; specializes in wood-firing harvested clay. (B) *Communion Light* and collection of vessels by Pilar Wiley; specializes in gourd-like forms and lighting fixtures. (C) *Sitting Dudes with Bowls* and *Reclining Dude on Blocks* by Eun-ha Paek; specializes in 3D-printed and hand-built figurative sculptures. (D) *Keishiki Clouds* and close-up on glaze work by Raina Lee; specializes in hand-built forms and textural glaze works. (E) *One Line Drawing* and *Goursat's Surface* by Camila Friedman-Gerlicz; specializes in sculptures inspired by 3D mathematical functions.

of fully simulating material behavior. Building effective software technologies for professional digital fabrication therefore requires a deep understanding of domain-specific materials and fabrication workflows to ensure that software operations align with professional manufacturing practices. In cases where software interfaces directly with complex material pipelines, the real-world needs and expertise of professional practitioners can provide crucial insights to guide development.

To explore these challenges, I participated in and helped facilitate two summer-long artist-in-residence programs titled Experimental Clay Residency Program[1]. These residencies brought together academic researchers and skilled ceramic craftspeople for extended collaboration and knowledge exchange. Over the course of the program, researchers and artists collaboratively developed software tools and produced thousands of physical artifacts, ranging from test pieces to works exhibited in galleries or sold to

[1] `https://handandmachine.org/index.php/experimental-clay-residency2`

customers. The collected data about resident and academic researcher accounts in the forms of structured interviews, discussions sessions, and written weekly reflections, and performed a reflexive qualitative analysis. This work and analysis foregrounded the complexities of bridging the research-practice divide in software development for physical production and led to the articulation of a novel approach: *practice-driven software development methodology* for digital fabrication [49]. The residencies demonstrated a systematic mechanism for achieving meaningful digital fabrication software contributions that mutually benefit artists and researchers.

The potential of a practice-driven approach becomes especially clear when examining the software tools developed during these residencies. Running a digital clay fabrication residency required the research team to build new tools tailored to the unique demands of clay fabrication, due to the relative absence of existing design software in this area. I discuss the ceramics manufacturing pipeline and some demands of clay's material agency in Section 2.3.2. This led to the creation of five software tools, ranging from low-level utilities to highly constrained, artifact-specific design systems. In this chapter, I present two prototype systems for computational clay fabrication, *ClayToolkit* in Section 4.2 and *CeramWrap* in Section 4.3, which originated in the context of the residencies [49, 50]. The other three tools, *CoilCAM* by Sam Bourgault *et al.* [160], *SketchPath* by Devon Frost *et al.* [161], and *ParaLight* by Jennifer Jacobs in [49], are not discussed further in this chapter. Together, the tools and their development methods offer a structured and replicable model for producing robust, novel computational fabrication systems. Figure 4.2 shows a timeline of these tools' development relative to the two residency periods to provide a broader perspective on the practice-driven software development for material production.

The research originates from a three-year long NSF project (Award # 2026286), in part, aiming to integrate long-standing craft practices with new digital fabrication

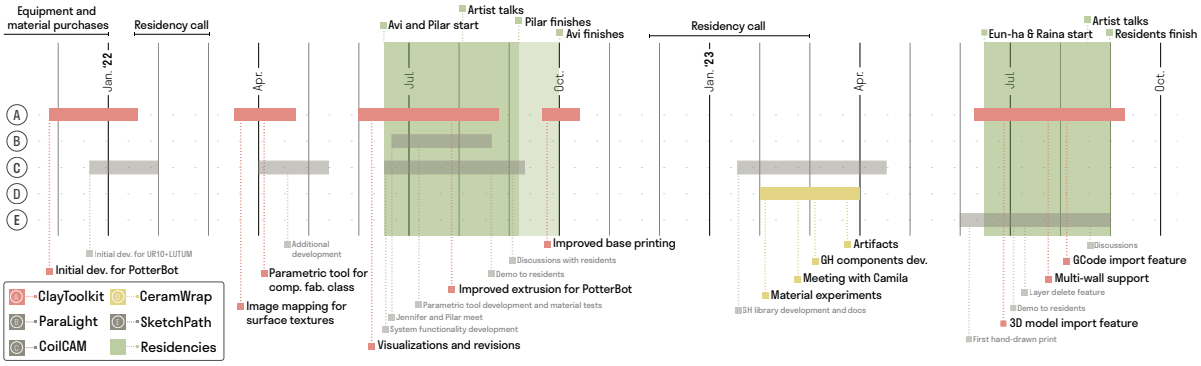


Figure 4.2: The timeline of software development activities before, during, and between our two consecutive residencies. We show the development periods of each software tool relative to the residency periods. Labels at the top of the diagram represent primary activities relating to the residencies, and labels at the bottom provide details of the development process behind each software. Dates where the research team worked together are highlighted in green in the summers of 2022 and 2023. I highlight the development activities regarding *ClayToolkit* (A) and *CeramWrap* (D) over other software.

technologies and to create new computational techniques inspired by craft traditions. The project was co-managed by Hands and Machine Lab[2] at University of New Mexico and Expressive Computation Lab[3] at UC Santa Barbara. A key component of the project was the Experimental Clay Residency Program, designed to foster deep collaboration between academic researchers and professional artists. While the residencies ran in parallel at UNM and UCSB, this chapter only discusses outcomes of the residencies at UCSB with participating artists Avi Farber, Pilar Wiley, Eun-ha Paek, and Raina Lee. One exception is Camila Friedman-Gerlicz– a UNM resident who periodically collaborated with the UCSB team. Figure 4.1 shows a selection of their works prior to the residencies. The goal of the residencies was to facilitate mutual knowledge exchange at the intersection of manual ceramic craft and computational fabrication. Parts of this chapter's content has been exhibited at an event and published as standalone HCI publications [49, 50]:

- Latentville, *Santa Barbara, CA*, 2023

[2] `https://handandmachine.org`
[3] `https://expressivecomputation.com`

- Toka*, Bourgault*, Friedman-Gerlicz, and Jacobs. 2023. *An Adaptable Workflow for Manual-Computational Ceramic Surface Ornamentation.* In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 24, 1-15 [50]

- Toka*, Frost*, Bourgault, Farber, Friedman-Gerlicz, Lee, Paek, Pilar Wiley, and Jacobs. 2024. *Practice-driven Software Development: A Collaborative Method for Digital Fabrication Systems Research in a Residency Program.* In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (DIS '24). Association for Computing Machinery, New York, NY, USA, 1192-1217 [49]

4.2 Case Study: *ClayToolkit*

The residency structure and the lack of existing clay 3D printing technologies required the development of low-level computational utilities and higher-level design tools to support the creative agency of the professional artists working with digital fabrication technologies. To address this, I created *ClayToolkit*, a CAM-based parametric design toolkit for creating forms, designing displacement-based surface details, and processing toolpaths [49].

I developed *ClayToolkit* continuously before, during, and between artist-in-residencies (Figure 4.2A). I compiled my initial experiments into a preliminary version of *ClayToolkit* as an educational tool for my teaching assistance to the computational fabrication class taught by Jennifer Jacobs in April 2022. Once the residencies have started in June 2022, I adapted this system for professional use and continued developing it for two summers based on resident's feedback. I fine-tuned the parametric design and low-level operations with the help of Avi and Pilar in 2022. I continued improving this tool with Eun-ha and Raina in 2023, and introduced two higher-level utilities for interfacing the realities of digital design and physical fabrication. Upon concluding the residencies, I released

ClayToolkit as an open-source toolkit[4].

4.2.1 System Design & Implementation

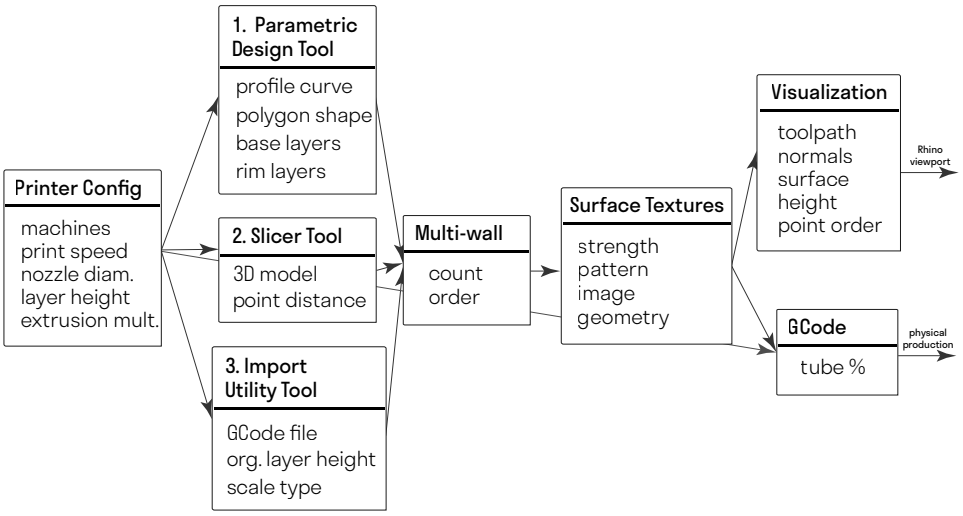


Figure 4.3: A simplified diagram showing *ClayToolkit*'s various components and their primary interactions.

ClayToolkit is a CAM-based design and utility toolkit to create and manipulate machine toolpaths for clay 3D printers. I implemented this system inside Rhino/Grasshopper [217] environment using custom Python scripts. It is designed to support artist's creative agency in (1) parametrically generating toolpaths that conform to their design intent, (2) manipulating toolpaths or existing solid geometry to be compatible with clay 3D printing. This is made possible by *ClayToolkit*'s three distinct sub-tools:

1. The *parametric design tool* supports designing radially symmetric toolpaths based on an input profile curve (Figure 4.4). The artist can also map predefined repetitive patterns, geometry, and images to add surface textures to the toolpath.

2. The *slicer tool* supports creating toolpaths from solid geometries and 3D models (Figure 4.5). The artist then might modulate the toolpath with images, other

[4]`https://github.com/merttoka/ClayToolkit`

geometry, or patterns to create surface textures. This functionality allows artists to manipulate geometry created through CAD or 3D scanning with lower-level machine operations of clay printers.

3. The *import utility tool* enables converting an existing GCode file saved with specific parameters back into an editable toolpath. This functionality provides interoperability between different software and machines.

Below, I explain the software implementation details of *ClayToolkit* and highlight the components for interfacing digital software with physical production requirements.

0. Shared Components

ClayToolkit is organized as a collection of Grasshopper modules that can be recombined for different workflows (Figure 4.3). Here I briefly explain the components shared between every design and utility tool:

Visualization module provides a preview of toolpaths as colored polylines (as shown in Figure 4.4B), normals as vectors, and vertices as an ordered and annotated point cloud within Rhino's 3D viewport. Additionally, it can create a colored solid geometry around the object to enhance the visibility of surface textures and display the vessel's height in millimeter within this viewport.

Printer configuration module contains detailed information like bed dimension and start/end GCodes about various clay printers, including the Tronxy Moore 2 Pro, LUTUM 5, Eazao, PotterBots Super 10, Micro, and Scara. It also includes other print parameters such as print speed, nozzle diameter, layer height, and extrusion multiplier.

Displacement-based surface texturing module comprises a configuration component that holds parameters and a custom Python component that modifies a given toolpath. The configuration allows modulating the toolpath using a pattern or an image

with a specified strength. The Python component then maps the selected modulation method to the toolpath vertices and modifies the vertices corresponding to the 2D masks based on their normals. A helper utility enables excluding specific regions from this surface texturing using Boolean geometry (e.g. the poodle's head in Figure 4.5B).

Multi-wall module enables offsetting walls and designating a print order to construct multiple shells around the object. The toolpath is not spiralized multiple walls is enabled. This feature addresses the challenge of inadequate support structures in clay printing. It facilitates the construction of stable prints, particularly when the print angle exceeds 45 degrees, and the wall ordering accommodates angles that widen or contract.

GCode export module, a fundamental component of *ClayToolkit*, enables the seamless integration of digital designs with physical machines. It takes all parameters specified in *ClayToolkit* as input, centers the object precisely within the printer bed's center, displays the tube usage, and stores these parameters as metadata. It then calculates the appropriate extrusion amount for each line segment of the toolpath (`l*layerToNozzleRatio+(layerToNozzleRatio+4/PI)`) and writes a list of machine instructions using `G1` GCode commands. The module inserts a `G92 E0` every 5m of cumulative clay extrusion to avoid numeric overflow. Once the file is constructed, it is saved to transfer to the machine's local storage.

1. The *Parametric Design Tool*

This design tool lets a craftsperson sketch a single profile curve and generate a radially-symmetric toolpath. It allows creating a parameterized polygonal or circular base. The component generates base and wall vertices sequentially. For each base layer, the Python component computes a double-spiral composed of `ceil(radius/lineSpacing)` many lines with `nozzleWidth+layerHeight` space between them. The first spiral moves towards inside at a fixed height, and the second grows outwards on the next layer height.

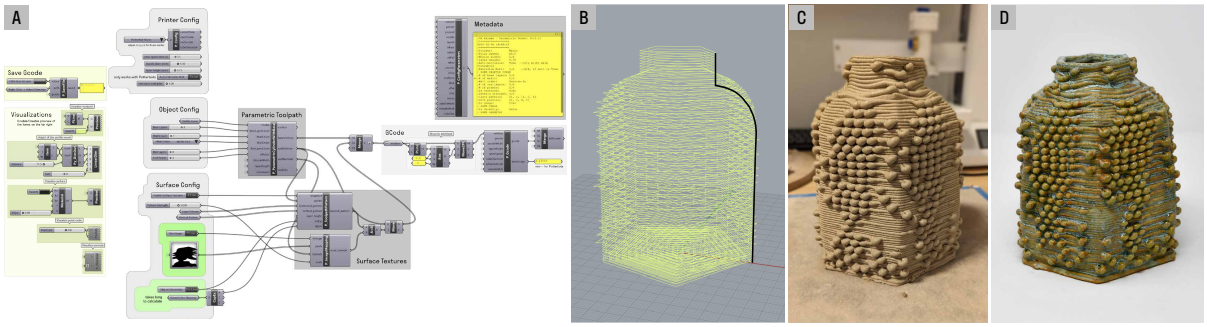


Figure 4.4: *ClayToolkit*'s primary design tool allows creating parametric toolpaths with surface textures. (A) The Grasshopper interface consists of visualization utilities, printer configurations, the object's form and surface configurations, parametric vessel generator, texturing module, and GCode export module. (B) The visualization of a toolpath generated by profile curve in bold. (C) 3D-printed wet object with a textured surface. (D) Finished ceramic object.

Wall vertices generation happens through a continuous spiral to avoid seems along one side of the vessel. The profile curve is sampled at each increment of the layer height, and the perpendicular distance between the sampled point and the world Z-axis determines vessel's radius at that point. Number of points along this lofted spiral layer is determined by the nozzle diameter, and the normals determined by the parameterized polygonal base shape. Finally, once the walls are generated, an optional layers are created at the rim of the object that repeats the latest layer multiple times. This was invented by one of the residents, Avi, aimed to produce a nicer finish along the 3D-printed rim.

Once the toolpath is complete, it then passes to the displacement-based surface texturing module to move vertices along their normal vectors based on the intended pattern. Figure 4.4B-D shows a combination of an image and pattern modifying the surface smooth to having surface bumps.

2. The *Slicer Tool*

This tool integrates traditional 3D-model/slicing workflow of FDM printing into clay fabrication workflows to add surface textures on the surface. It targets artists who already

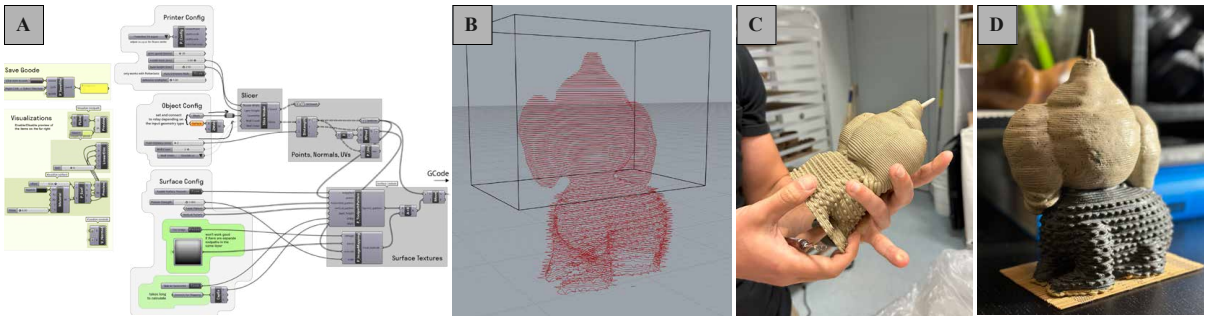


Figure 4.5: *ClayToolkit*'s slicer tool enables generating toolpaths from 3D models and allows modifications with other operations in the toolkit. (A) The Grasshopper interface consists of (left to right) software visualizations, printer input configurations, the object and its surface, the slicer, and the texturing module. (B) Eun-ha's signature poodle model converted into a toolpath with surface textures. Head vertices (inside the box) are excluded from texturing. (C) Trimming the leather hard object. (D) Glazed ceramic object, *Poodle with a Sweater*.

have a watertight 3D model (e.g. designed in Blender) and need further clay fabrication-specific modifications. Figure 4.5 exemplifies this need by showing Eun-ha's signature poodle model sliced and modified with selective surface texturing.

A dedicated Python script creates a sequences of XY planes translated by the layer height, and calculates the intersection points between the mesh and the planes based on the point distance variable. In the case where the intersections produce multiple closed curves, the toolpath connects between these closed shapes, and nozzle extrudes in between these shapes during the print. The craftsperson is free to remove these bridging portions during or after the print is complete. This tool does not produce base vertices. Once the toolpath is created, the same surface texturing module described above applies to its path.

3. The *Import Utility Tool*

Traditionally, GCode files are reserved for executing machine-instructions and are not considered a part of a design workflow. However, throughout the residencies, various design tools and machines were employed, and files created in one system with certain

parameters couldn't be modified in other software or run in different machines. To address this issue, I implemented a third tool in *ClayToolkit* to read a GCode file and recreate its path as an editable toolpath in Grasshopper environment.

This import utility allows craftsperson to import an existing GCode file, and re-parameterize it inside Grasshopper. This is done by parsing the `G1` lines, converting each `XYZ` pair into a Grasshopper point, and piping these points in similar modules describes above. If the toolpath is spiralized, it requires a user input for the original layer height to decide how to modify the shape. When the new layer height is different than the original, the toolpath gets scaled either uniformly or only on Z-axis to match this target layer height. Upon successful recreation of the toolpath, surface texturing or multi-wall operations can be applied, and the GCode export module creates the machine-readable file with the new printer configurations.

4.2.2 Physical Production

ClayToolkit were primarily used with 3DPotter's Potterbot machines [142], particularly the smaller *PotterBot Micro* and the larger *PotterBot Super 10* (Figure 4.6D). Observing residents' daily physical struggles and hearing them voice their fatigue motivated me to prioritize software solutions that could ease physical labor.

Calibrating Extrusion for PotterBot: My and residents' experiences with PotterBots began with frustration. The printed layers separated due to unpredictable extrusion flow, and finding a suitable extrusion multiplier on the Duet interface while printing was challenging. Troubleshooting the pipeline revealed that extrusion commands for each line segment (`E...` values) did not depend on nozzle diameter. I hypothesized that the volume of clay flowing through one nozzle must differ from another nozzle.

I designed an experiment to measure how PotterBot 10 Super responds to GCode

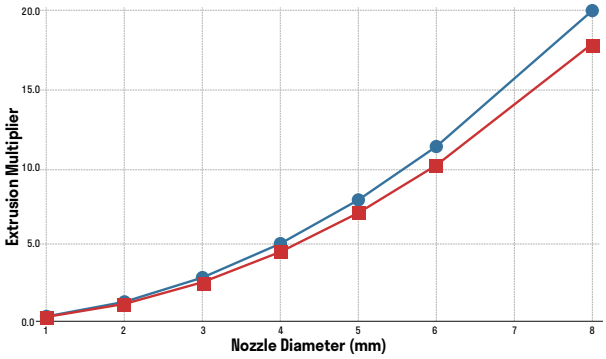


Figure 4.6: Tube loading workflow for PotterBot Super 10. (A) Eighteen pounds of clay gets wedged and shaped into a tube-sized cylinder. (B) Wedged clay gets loaded into the wall-mounted extruder and compressed into the PotterBot tube. (C) Sixteen screws are screwed to attach the motor to the top of the tube and the nozzle to the bottom. (D) The tube gets hefted into the PotterBot's z-axis clamp, and the bed is prepped.

extrusion commands. Marking piston's location with tape in an empty tube and running `G1 E5000` moved the it down by 2 mm, `G1 E10000` moved it by 4 mm. I repeated this step many times to account for precision errors. The volume of displaced material under the piston must be equal to material volume extruded through the nozzle: $V_{\text{displacedClay}} = \pi(d_{\text{piston}}/2)^2 * l_{\text{pistonMove}} = \pi(d_{\text{nozzle}}/2)^2 * l_{\text{filament}}$, where d denotes diameter and l denotes length. Inserted the experiment values (Figure 4.7-Left) into this equation and solving for l_{filament} gave me the hypothetical length of the extruded clay filament, and dividing 5000 by this number gives us the extrusion multipliers per nozzle diameter. As shown in Figure 4.7, these values scale non-linearly as the nozzle grows larger.

I then integrated this calculation into *ClayToolkit*'s GCode generator and observed calibrated clay flow with every nozzle diameter. Occasionally we needed to adjust the rates using Duet controller due to clay consistency, however these adjustments were in the range of +-30% and made finding the comfortable range much easier. In addition, I implemented a *tube usage calculator* in *ClayToolkit*, allowing artists to view the percentage of tube required to print a given form before exporting the GCode. This functionality

	Super 10	Micro
Piston Diameter (mm)	95	70
G1 E5000 Piston Movement (mm)	2	3.29
Usable Tube Height (mm)	430	270



Nozzle Diameter (mm)	Extrusion Multiplier (Super 10 ■)	Extrusion Multiplier (Micro ●)
1	0.2770	0.3102
2	1.1080	1.2406
3	2.4931	2.7914
4	4.4321	4.9625
5	6.9252	7.7539
6	9.9723	11.1656
8	17.7285	19.8499

Figure 4.7: Data and result of the experiments for calibrating Potterbot Super 10 and Micro extrusion. (Left) Measurement values; (Middle) a plot of extrusion multiplier values for Super 10 ■ and Micro ●; (Right) data table of the same plot.

enabled residents to plan their clay usage and determine if they would need to execute a tube swap mid-print. It shaped Raina's aesthetic decisions: "*I kept looking at the clay usage, and if something had too much texture, I would ease up on the texture and then just print it.*"

ClayToolkit introduced a collection of tools for designing and manipulating toolpaths for clay 3D printing. The available methods in this toolkit collectively allow for interacting with material agencies using software and digital fabrication technologies. This toolkit primarily allows for designing the *overall form* of an object and helps adding *displacement-based surface details*. *ClayToolkit* played a crucial role in professional artist's ceramic production due to its flexible features and abstracting machine instructions while keeping expressive potential. It provided vital utilities to fill the gaps in the sparse domain of existing clay 3D printing design technologies. It also acted as a repository for the growing knowledge about clay 3D printing techniques, manifested as discrete computational design and processing methods.

In the next section, I present another case study, *CeramWrap*, to specifically target the space of ceramic *surface decorations*. This case study introduces a novel manual-

computational texturing method for ceramic objects, and integrates the agency of materials into an adaptable and non-linear workflow that can function with various fabrication methods and decorative materials.

4.3 Case Study: *CeramWrap*

The displacement-based surface texture method described in *ClayToolkit* uses unique affordances of digital clay fabrication, due to its precision and granular control. These textures can create woven-like qualities on vessel surfaces (e.g. Figure 4.4C-D), and it is generally incredibly laborious to recreate manually. However, the space of ceramic surface decorations is bigger than displacement-based methods, and also contains diverse additive or subtractive techniques and glaze materials. This variety makes surface decorations a good candidate for studying material and human agencies.

An analysis of professional ceramics manufacture identified four key aesthetic dimensions for surface decoration: *the type of decorative elements* (e.g. single motifs vs. patterns), *their positioning* (local vs. global), *their alignment to grid structures* (from strict to freeform), and *their relationship to vessel geometry* (general-purpose vs. form-specific). Manual techniques allow for precise, localized application and geometric adaptation, whereas patterned methods offer efficiency but less flexibility. Ornamentation that conforms to 3D vessel forms often sacrifices repeatability, while repeatable methods struggle with shape specificity. This tension points to the potential of computational approaches that support both flexible and systematic applications. This created an opportunity to develop a manual-computational workflow that can support both single elements and patterns and use local and global positioning techniques that may be grid-based or grid-free while providing vessel-specific capabilities.

CeramWrap [50] is an interactive design workflow that lets craftspeople to design

and fabricate decorative patterned stencils tailored to radially symmetric vessels. Rather than presenting one rigid and linear technique, *CeramWrap* embraces the iterative and often cyclical nature of material production in an effort to augment established skilled ceramic ornamentation in a non-linear manner. This section describes the adaptable workflow steps and the computational tools developed to bridge digital and physical design domains.

CeramWrap was conceived in a collaboration between three HCI researchers and a professional ceramicist after the first residency (Figure 4.2D). Sam Bourgault and I are Ph.D. researchers working at the intersection of media art and HCI, developing software and techniques for digital and physical workflows. Camila Friedman-Gerlicz is a professional ceramic artist and HCI researcher who creates sculptures inspired by 3D mathematical functions. Her work includes extensive use of manual and CAD-based methods and often involves the creation of complex slab-building templates through the unrolling of 3D virtual models. Jennifer Jacobs is an assistant professor and director of Expressive Computation Lab who designs and researches computational tools informed by art and craft practice. During the development of the project, Camila provided crucial insights on surface unrolling and its connection to ceramic applications. Sam and I developed the computational steps of *CeramWrap*'s workflow. Sam implemented the procedural pattern generator, and I developed the computational surface unroller that bridges the digital-physical spaces. I performed extensive material and application tests with digitally-unrolled and fabricated stencils. Sam, Jennifer, and I experimented with various manual application techniques and materials.

4.3.1 Adaptable Workflow Development

CeramWrap workflow consists of four stages (highlighted in blue in Figure 4.8):

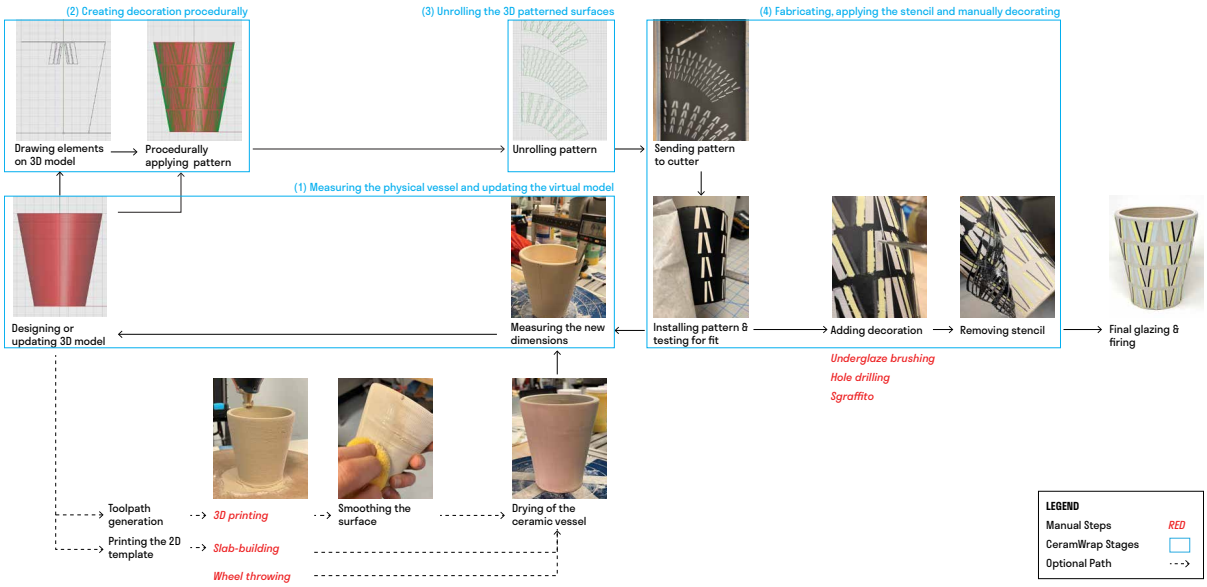


Figure 4.8: The interactive workflow of *CeramWrap* aligns with ceramic craft practices. The flowchart shows a simplified version of our workflow, and images represent the production steps of one of our examples (a variation of Figure 4.16C). The steps framed by blue rectangles refer to the four stages of our workflow.

W1. Measuring and averaging physical vessel dimensions and creating or updating a digital vessel model (described in Section 4.3.1).

W2. Decorating the digital model through procedural pattern generation (described in Section 4.3.2).

W3. Unrolling the 3D patterned surfaces of the vessel into 2D surfaces suitable for fabrication (described in Section 4.3.2).

W4. Fabricating and applying the stencil to the physical vessel for manual ornamentation (described in Section 4.3.3).

This workflow integrates computational techniques with existing ceramic practices of surface decoration. As highlighted in red in Figure 4.8, *CeramWrap* can be used with different methods for form building (e.g. slab building, clay 3D printing, and wheel throwing) and surface decorations (e.g. underglaze brushing, subtractive drilling and carving,

and sgraffito). I provide details of the workflow steps distributed across Sections 4.3.1, 4.3.2, and 4.3.3 to highlight their primary domains of operation, however these steps can be used in a *non-linear* fashion and backtrack to previous steps in the workflow. This is due to the complexity of ceramic manufacture and the computational workflows must support this complexity by developing adaptable methods. The first step (**W1.**) requires iteratively moving back and forth between digital and physical domains, and described below.

W1. Measuring the Physical Vessel and Updating the Virtual Model

The first stage of this workflow begins after a craftsperson has fabricated their vessel either through manual methods (wheel throwing or slab building) or digital fabrication (clay 3D printing). They then measure the dimensions of this physical vessel at a phase suitable for the chosen decoration technique and create or update the virtual model based on these measurements. We presently assume the digital model is generated using simple 3D modeling functions such as the loft function that creates a 3D surface by connecting two or more closed curves or the revolve function by moving a profile curve around a closed curve. As a result, *CeramWrap* is limited to radially symmetrical surfaces.

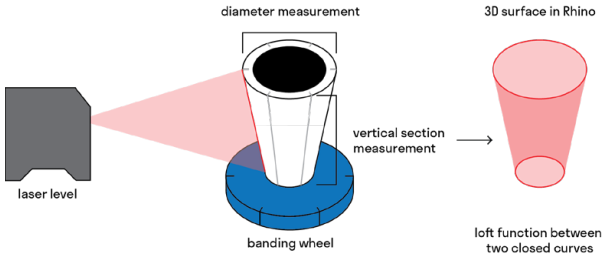


Figure 4.9: Marking the vessel for measurements. The banding wheel and the laser level are used to divide the vessel into six sections with vertical markings. Three diameter and six vertical measurements are taken with calipers and averaged to create a virtual model of the vessel's surface using simple modeling functions available in the Rhino software.

As described in Section 2.3.2, clay undergoes various transformations before becoming

ceramic that impact the dimensions and form of a vessel. Its dimensions at the time of surface decoration application will be different from the original dimensions. Therefore, measuring is a critical part of the workflow because each of *CeramWrap*'s following stages will be executed with respect to the physical vessel's dimensions at the time of surface decorations.

To make precise and consistent measurements, the craftsperson can use established ceramics measurement techniques. In this workflow, they divide the physical object's base circumference into six equal regions using the grooves on a banding wheel. They expand these points to the side of the vessel with a laser level. The craftsperson then marks the critical points for digital reconstruction along the laser level. Using these markings as guides, the craftsperson takes three diameter measurements at different heights and six vertical measurements along the walls of the vessel. The craftsperson then uses the averaged dimensions to generate or update a radially symmetrical digital model. The measurements are taken on the outer surface of the vessel to account for the material thickness. This procedure is illustrated in Figure 4.9 with a tapered cylinder.

Depending on the precision required by the decorative patterns, the craftsperson may repeat the measurement stage multiple times until the digital model represents the physical object adequately. *CeramWrap* automatically propagates any updates to the digital model to the following stages of the workflow, enabling the craftsperson to generate, fabricate and test the fit of the template on the physical vessel quickly.

This stage of the workflow is beneficial for two reasons. First, it ensures that the unpredictability of a drying ceramic vessel is represented as accurately as needed by the design intend. Second, it opens up the workflow's capabilities to not only clay 3D printing but also to traditional methods of form building in ceramics, such as wheel throwing and slab building.

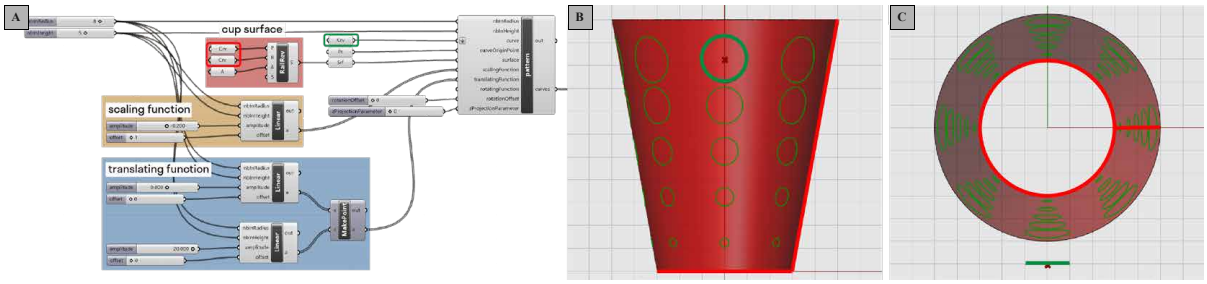


Figure 4.10: *CeramWrap*'s custom Python component to create procedural patterns on digital surfaces. (A) The Grasshopper program generates eight columns by five rows of circles on the surface of a tapered cylinder. (B) A side and (C) top view of the generated pattern on the cup. The curves used to create the cup shape are highlighted in red, and the curve used for the initial circle element is green.

4.3.2 Digital System Implementation

Computational tools of *CeramWrap* workflow are developed using the Rhino and Grasshopper environments, which supports the 3D modeling of digital vessels and the parametric design of decorative patterns. Second (W2.) and third (W3.) steps of the workflow shown in Figure 4.8 are described below:

W2. Procedurally Generating Patterns on 3D Model

Ceramicists use manual tools to create grid-based designs on ceramic vessels [218] or non-parametric tools to generate fitting stencils [219]. These approaches are challenging to reproduce consistently or evaluate aesthetically before physical application. Mathematically defined procedural design strategies can support rapid instantiation of visual elements in a grid structure and visualization onto 3D digital models.

To support these procedural strategies, *CeramWrap*'s workflow includes a custom component that enables craftspeople to create and display patterns on virtual ceramic models. Craftspeople draw decorative elements in the model's front face using Rhino's 2D drawing functions. They then input these elements and the 3D model created in the previous workflow stage to the pattern script. Craftspeople specify the horizontal and

vertical repetition parameters for the element's new instances and how these instances should scale and translate across the surface. This results in decorative instances propagating across the virtual model's surface. In Figure 4.10, a linear function is used as the `scalingFunction` parameter to reduce circles' radii towards the bottom, and a 2D linear function is used as the `translatingFunction` parameter to distribute the circles' center at equal distances. Craftspeople can also use the `rotationOffset` parameter to move the entire pattern around the model to optimize its location relative to the stencil's creases created in the following stage.

This approach enables both the rapid prototyping of grid-based patterns directly on 3D surfaces and the customization of each element's scale, position, and rotation. It also generates as output the necessary curves used by the unrolling stage described below.

W3. Unrolling 3D Patterned Surfaces into 2D Surfaces

The third stage of the workflow consists of unrolling the 3D pattern from the previous stage into a 2D surface. Ideally, the unrolling results in a one-to-one mapping between the 3D points on the virtual model's surface and the generated 2D surface, but, in general, perfect mapping is not possible. However, a particular class of mathematical surfaces, *developable surfaces*, allow for a perfect match between smooth digital surfaces with zero Gaussian curvature and unrolled geometry suitable for fabrication [220, 221]. Any geometry with planar surfaces (e.g. a *polygon* extruded along a *line*) and single curvature (e.g. a *circle* extruded along a *line*) are examples of developable surfaces. When a model contains double-curvature surfaces (e.g. a *circle* extruded along a *curve*) there is no one-to-one mapping between a 3D and a 2D surface suitable for fabrication. Unrolling this type of surface requires discretizing one of the curvatures and creating an approximation of the 3D geometry. In Rhino, this simplification equates to discretizing one of the rail curves used to create the 3D model into a polyline with a variable number

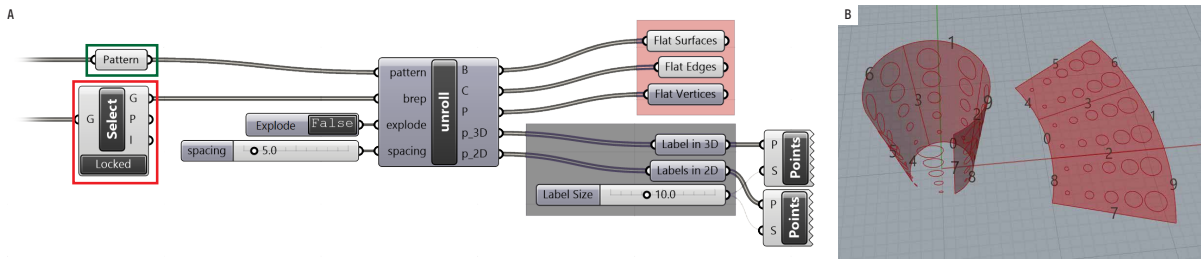


Figure 4.11: Interactively unrolling the 3D patterned surface. (A) The Grasshopper program unrolls the selected faces of a tapered cylinder (in red) with the 3D pattern from Figure 4.10 (in green). (B) A perspective view of the selected 3D surfaces and corresponding 2D unrolled surfaces, along with patterns and labels for reference.

of edges. The approximated single-curvature surface can then be unrolled into a planar surface suitable for fabrication. Since the mapping between the original 3D model and the approximated 2D surfaces is not preserved, in practice, we avoid placing decorative elements at the edges of the approximated 3D vessel.

For unrolling the 3D patterned surfaces created in the previous stage into 2D surfaces, I implemented a Python component based on the unroll function in Rhino API as shown in Figure 4.11. The craftsperson uses the decorative elements' curves on the 3D surface from the previous stage (`pattern` parameter) and the 3D model surface of the vessel (`brep` parameter) as input to this script. In general, the script outputs a flat representation of the patterned surfaces composing the 3D model, where the 2D surfaces are connected in the exact order of their arrangement within the model (Figure 4.11B). The unrolling script may yield intersecting 2D surfaces unsuitable for fabrication, especially in vessels with numerous surfaces of varying curvatures. To address this issue, the craftsperson can use the `explode` and `spacing` parameters to separate the generated 2D surfaces with a specified spacing offset. By doing so, the connections between the sides of the surfaces are lost, but the resulting geometries can be fabricated. Exploding the generated 2D surfaces is also necessary when working with a double-curvature vessel model that has been simplified along one of its dimensions since the 2D surfaces will not match perfectly

the original model.

The craftsperson can also unroll regions of interest on the vessel geometry selectively using the Selectable Preview plugin [222] framed by a red rectangle in Figure 4.11A. In Figure 4.11B, we show the interactive selection of three quadrants of the tapered cylinder and its corresponding unrolled surface. This selective unrolling facilitates the design of horizontal and vertical patterned strips across a vessel. Figure 4.12 shows the application of selective unrolling on the neck and the middle section of the vessel. We unrolled the neck section in one piece because of the simplicity of the 2D generated surface (Figure 4.12B) and we unrolled the bottom bulbous section in separate pieces to facilitate the installation of the stencil on the surface.

CeramWrap's unroll script also enables labeling the edges of the unrolled geometry and the 3D form in digital space using `p_3D` and `p_2D` output parameters. By turning the preview of the `Points` components on and adjusting the `Label Size` parameter in Figure 4.11A, the craftsperson can visualize digital labels for corresponding edges. Once the 2D patterned surfaces are generated, the craftsperson exports the results in a machine-readable vector format using Rhino's export functionalities. The craftsperson may or may not choose to keep the digital labels on the screen before exporting, and these labels would be fabricated along with the stencil if left on.

The interactive surface unrolling and procedural pattern generation methods provide multiple benefits for creating custom surface decorations for ceramic vessels. The workflow supports selective unrolling of a region of interest on the vessel geometry, which aligns with existing ceramic decoration practice that focuses on the design of vertical and horizontal patterns as illustrated in our design space. In cases where the 3D model contains double-curvature regions, interactively designing patterns on the 3D surface helps reduce issues arising from the approximated unrolled geometry and the physical object.

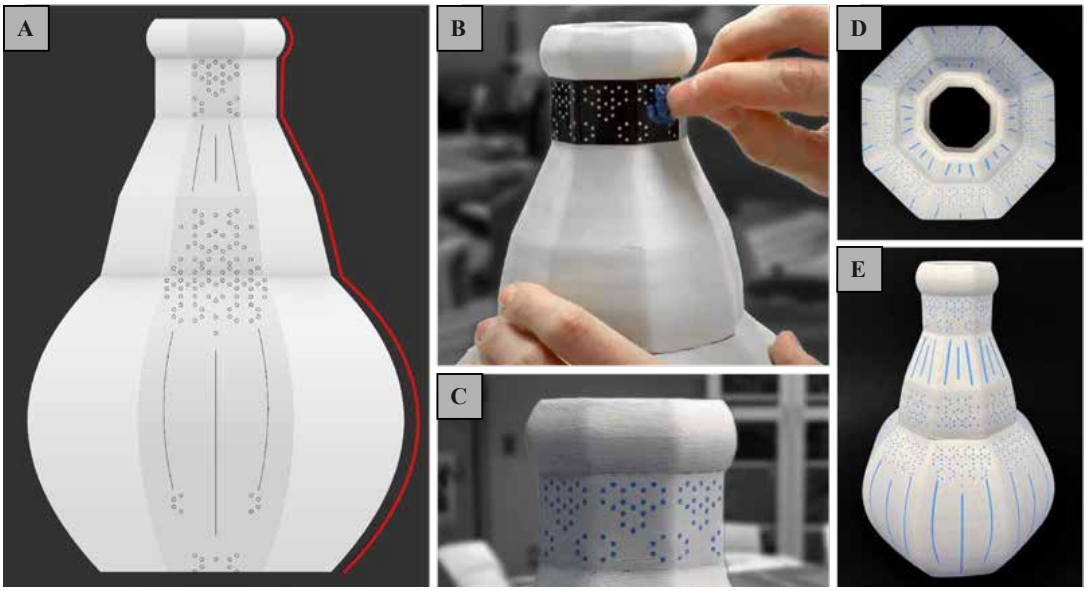


Figure 4.12: *CeramWrap* example workflow. (A) The digital model of a 3D-printed vase with decorative patterns. (B) The craftsperson installs the vinyl-cut stencil on the neck after selectively unrolling its surface and applies blue underglaze. (C) The resulting pattern after the stencil is removed. (D) A top and (E) a side view of the final vessel.

4.3.3 Physical Production

Once the design is finished and unrolled, the fourth stage of the workflow (Figure 4.8-4) consists of fabricating the unrolled stencil, applying it on the surface of the physical vessel, and manually decorating the vessel with a chosen method. This stage involves the possibility of going back to the measuring stage (W1.) if the stencil does not fit the physical vessel adequately. The entire workflow is built to support these discrepancies and facilitates a rapid update of the stencil.

W4. Fabricating and Applying the Transfer Geometry

This fourth stage of the workflow requires a high level of craftsmanship due to involves the stencil alignment on the physical vessel and manual decoration. Therefore it necessitates a good understanding of the materials and decoration techniques in use.

During this stage, the craftsperson cuts their patterned stencils in the chosen material (e.g. newsprint, paper, or vinyl) on a digital fabrication machine (e.g. a laser cutter or a vinyl cutter). We found that newsprint paper works well with objects in the leather-hard phase, standard linen or copier paper in the leather-hard and bone-dry phases, and adhesive vinyl in the bone-dry or bisque-fired phases. The craftsperson should therefore choose the stencil material that best fits the decorative technique they want to use based on the required clay phase for that technique.

Once fabricated, the craftsperson applies the stencil to the clay surface. To apply a newsprint stencil on greenware, they can use water to create adhesion between the clay and the paper while avoiding excessive moisture to maintain the structural integrity of the vessel. After aligning the newsprint stencil with the vessel, additional water is applied to adhere the stencil to the surface. The craftsperson can then apply their desired surface decoration medium through the stencil's openings. This time-sensitive approach requires relatively fast decoration application since evaporation will de-adhere the newsprint and extra water will weaken the greenware.

To apply a vinyl stencil on a bone-dry or bisque-fired surface, the craftsperson must first weed the vinyl using tweezers. Then, using a transfer tape, they release the stencil from the liner material, align the transferred vinyl with the ceramic surface, and apply force to ensure good adhesion while avoiding buckling in the stencil. After carefully removing the transfer tape, the craftsperson applies the decoration medium like glaze or underglaze, lets it dry, and removes the vinyl carefully. Stage four of Figure 4.8 demonstrates the fabricated 2D vinyl stencil, its installation on the vessel, the decoration through underglaze brushing, and the removal processes.

Lastly, a paper stencil can be used for applications where the bleeding of the decoration medium is not a concern, such as for the sgraffito technique. In this case, the craftsperson aligns and tapes the paper on the leather-hard or bone-dry vessel. Then

they use sharp clay tools to carve the outer surface of the vessel.

The choice of the stencil material impacts the surface decoration process. Laser-cut newsprint presents challenges for fabricating patterns inside hollow regions, and fine spacing between decorative elements complicates alignment and installation on ceramic vessels. Vinyl-cut stencils with transfer tape address these challenges but take longer to install on the clay vessel.

CeramWrap's generation of 2D patterned surfaces in vector format supports multiple stencil fabrication materials and techniques. The stencils generated through our workflow can therefore be customized to fulfill the requirements of the craftsperson's desired surface decoration method. The iterative process which takes place through the four stages of the workflow also supports rapid updates of the stencil to achieve a desired fit with the ceramic vessel.

4.4 Results

The two case studies presented in Sections 4.2 and 4.3 covered methods for clay fabrication with respect to creating digital forms, material influences on the fabricability of forms, and their surface details and decorations. In this section, I present outcomes produced by these methods by highlighting the design decisions they support.

4.4.1 Material Behavior Experiments

Clay displays hard-to-predict material behaviors due to its ever-changing physical state (i.e. its plasticity and moisture) throughout its production pipeline (its chemical transformation in kiln firing discussed in Section 5.4). In digital fabrication domains with uniform and stable materials like PLA, digital simulation can provide a partial means to predict artifact performance in relation to material properties; however with clay,

adaptable material engagement approaches are required.

Right after the fabrication, wet clay is easily deformed due to its extreme plasticity. While this is an important design affordance that allows artists to intentionally work on the form, it also unintentionally impacts the form's geometry through steps taken in the fabrication process. For example, wheel-thrown vessels are deformed when removed from the bat, and 3D-printed vessels are frequently deformed around the under-supported regions. Leather-hard objects are less subject to severe deformation, yet any action taken on the vessel is reflected in the final vitrified ceramic. Craftspeople often further modify or refine vessels in the leather-hard phase to avoid unintentional deformations through intentional actions like trimming, carving, and drilling. The vessel stops being deformable when it reaches the bone-dry phase. Craftspeople may make minor alterations like sanding or polishing, however, many actions are infeasible because the vessel becomes extremely fragile.

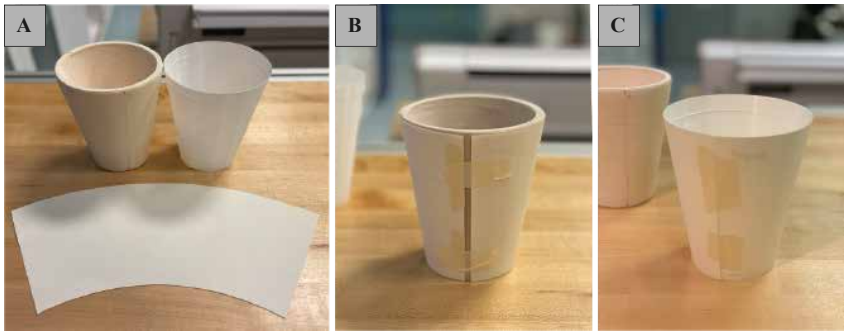
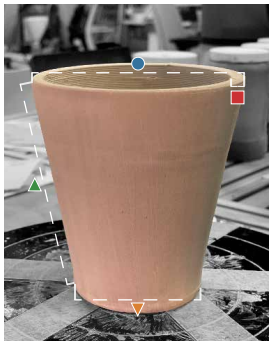


Figure 4.13: An experiment demonstrating the deformations on ceramic objects: (A) Clay and PLA objects and the unrolled surface fabricated in paper; PLA is fabricated using a second 3D model based on the outer diameter of the post-fired clay object. (B) Applying the template to clay object shows a loose fit due to deformations. (C) The same template aligns nicely on PLA object's surface.

It is generally infeasible to predict deformations since they originate from complex interactions between the material, object geometry, craftsperson, and environment. This is true for both manual ceramics and clay 3D printing. Figure 4.13 shows an initial test that revealed the degree of deformation on a simple clay 3D-printed tapered cylinder. We

3D printed a clay cylinder, removed it from the bat, and bisque fired it. We measured and recreated the digital vessel as explained in *CeramWrap* in Section 4.3.1, fabricated the unrolled surface of the recreated vessel with paper but the paper did not conform to the ceramics piece neatly. To confirm the accuracy of the unroll method (explained in Section 4.3.2), we 3D printed the recreated digital model in PLA using an Ender 3 Pro. The paper template fits perfectly with PLA object. This simple test shows how clay deformations can impact the development of precise methods that marries digital and physical approaches and reinforces the need for adaptable workflows in addressing digital clay fabrication complexities.

In addition to surface deformations, clay objects also shrink due to evaporating water content or chemical transformation. Shrinkage rates are dependent on the material composition, geometry, and dimensions. Although clay manufacturers provide an estimated shrinkage rate between wet clay and glaze firing (e.g. Aardvark[5]'s Nara 5 has a predicted final shrinkage rate of 12%), we observed shrinkage is affected by vessel geometry. Differ-



	Wet	Leather hard	Bone dry	Bisque fired	Glaze fired
Rim Diameter	103.5	98.5	97.9	96	91.54
Base Diameter	67.7	63.3	61.8	60.8	58.88
Profile Height	114	109	106.5	103.4	97.49
Wall Thickness	6	5.64	5.35	5.27	5.1

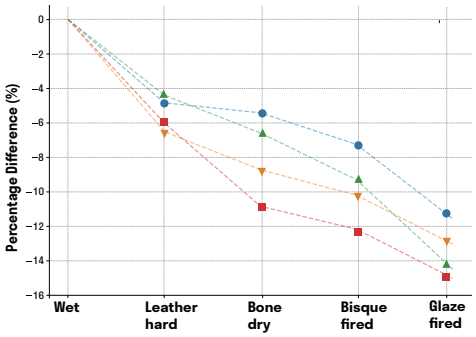


Figure 4.14: Clay shrinks non-uniformly depending on the vessel geometry: (Left) A photo of a tapered cylinder printed with Nara 5 and annotated with four measurements: ● Rim Diameter, ▼ Base Diameter, ▲ Profile Height, and ■ Wall Thickness. (Middle) A table showing the same measurements in mm at five phases of clay. (Right) A plot of the same data points visualized as a percentage change from the wet measurements.

ent parts of an object dries at different rates, and observing non-uniform shrinkage across

[5]A well-known clay retail store in California.

different parts of the same vessel is common. Figure 4.14 reports four measurements of the tapered cylinder on the left at five phases of clay and plots the percentage differences. Craftspeople accommodate shrinkage in their design process by fabricating vessels with consistent wall thickness. They also dry pieces in a controlled manner to avoid cracking by frequently checking the dryness across different parts to adjust air flow.

I discuss these experiments to demonstrate how clay behavior throughout the production pipeline affects new technologies if the application requires precise control. Adaptable and non-linear workflow, as shown in *CeramWrap*, can overcome these factors with digital design and rapid fabrication technologies. In other expressive technologies that does not rely on precise applications of various materials, like in *ClayToolkit*, the root causes of these factors (i.e. clay state and chemical transformations) shows up as other forms of unpredictable behaviors, discussed in Section 5.4.

4.4.2 Surface Detailing Approaches by System

Systems described in this chapter can produce intricate details on the surfaces of the geometries. I divide these into two, *displacement-based* and *ornamentation* methods.

Displacement-based Surface Texturing in *ClayToolkit*

This type of surface treatment method is common in clay 3D printing. The CAM-based toolpath is modulated based on (1) a repeating pattern sequence or (2) an image texture.

The patterns are defined by two sequences of floating point values for horizontal (H) and vertical (V) axes. For instance, the left cup in Figure 4.15A uses `V=(0, 1)` to skip texturing one layer completely and `H=(0, 0, 1)` to displace one in every three vertices in a single layer. When the number of vertices in a layer is not divisible by the number

of items in the pattern sequence, the left over index is carried over to the next layer and may create spiraling textures visible in the right cup on the right in Figure 4.15.

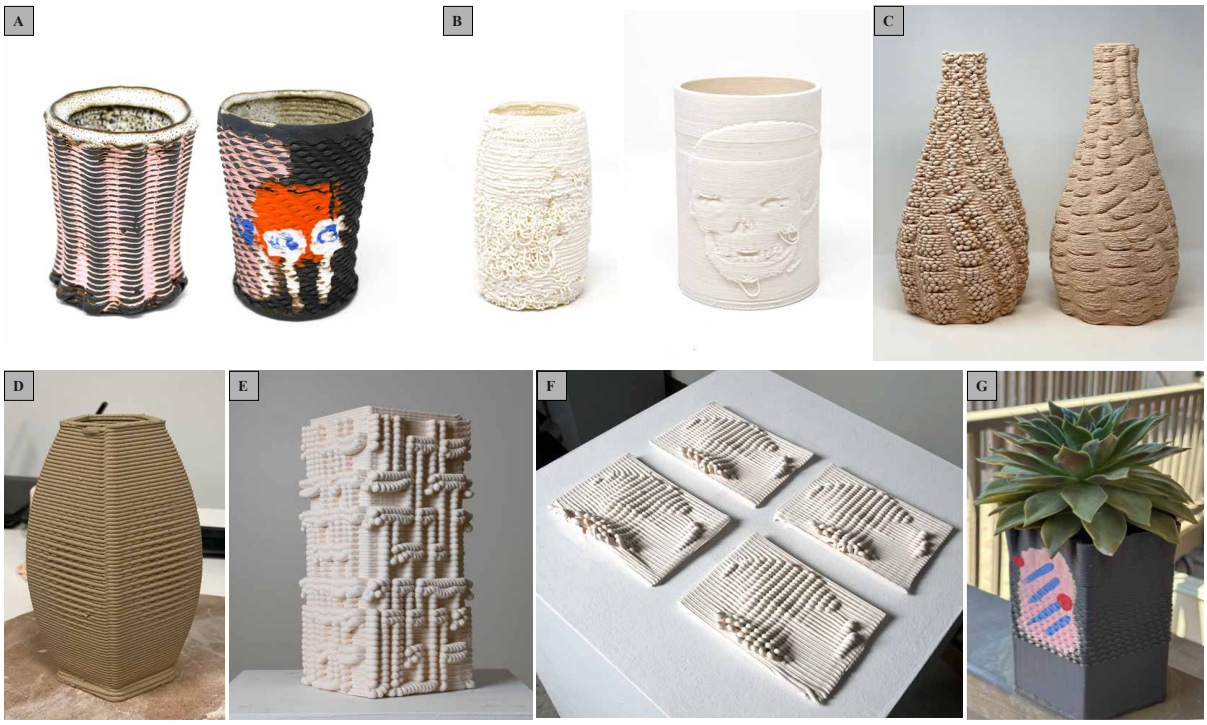


Figure 4.15: Various examples of artifacts using displacement-based surface textures in *ClayToolkit*. (A) Avi's cups showing patterns at regular intervals. (B) Eun-ha's porcelain objects with image textures. (C) Large vases with a generative image applied; left shows modulating the strength of a pattern, the right maps the image directly. (D) A test print with negative image texture strength. (E) Raina's image–textured cube tower. (F) Raina applied image textures on a cube and cut the sides to create flat relieves. (G) A hexagonal planter with gradually decreasing surface patterns.

The vertices are displaced along the normal directions, and pattern value multiplied by global parameter pattern strength describes the displacement distance for each vertex. When the displacement distance is too large, clay droops on to the surface as shown in Figure 4.15B-left. Using negative values for the strength or pattern indices is possible, and reverses the displacement direction to inside the object (Figure 4.15D).

Image textures behave in a similar way, each pattern value is sampled from a UV-mapping between toolpath vertices and the image pixel data (Figures 4.15E and B-right).

When image mapping and pattern sequences are used at the same time, image data still modulates the displacement strength, but pattern values of 0 cancels the displacing effect. For example, Figure 4.15C demonstrates this difference side-by-side: the left vase uses `H=(0, 1)/V=(0, 1)` pattern sequence and a generative image, and the right vase only uses the image data. I used this effect to create gradients in pattern sequence strengths as shown in Figures 4.15G and 4.23E.

Figure 4.15F shows an unexpected workflow using this method. Raina experimented with printing flat relief portraits using clay 3D printing, however, after realizing that horizontal prints yields weak objects, she decided to print an extruded cube and apply relief textures on its surface. She later cut the sides of this cube while leather-hard and fired the pieces separately. Figures 4.21, 4.23, 4.24, 4.26-left, and 4.27 show other displacement-based texturing results.

Surface Ornamentation in *CeramWrap*

In contrast to displacement-based methods unique to clay 3D printing, *CeramWrap* interactive workflow allows for producing surface ornamentation closer to how ceramicists decorate their artifacts. Seven example vessels shown in Figure 4.16 demonstrate how *CeramWrap* enables the computational design and fabrication of ornamentation on vessels with planar, single, and double-curvature surfaces, fabricated with manual methods or clay 3D printing. We use the workflow with underglaze brushing, drill piercing, and sgraffito surface decoration techniques.

(A) Drill-piercing on Double-curvature Surface on Wheel-thrown Colander:

We created a double-curvature vessel by wheel throwing (Figure 4.17A), and made a digital version of its leather-hard state. After approximating the double-curvature surface with ten vertical sections, we created a drill-pattern with 3/32" and 3/16"-diameter holes

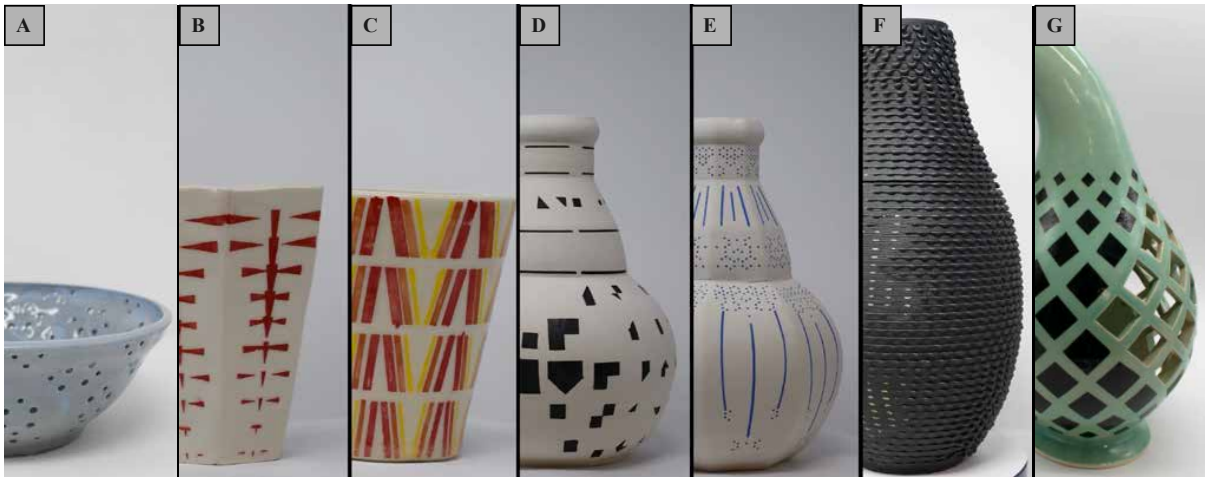


Figure 4.16: Seven artifacts decorated with *CeramWrap* workflow: (A) Colander by Sam and Jennifer, (B) hexagonal cup by Sam and I, (C) cylindrical cup by Sam and I, (D) bulbous vase by Sam and I, (E) octagonal vase by Sam and I, (F) large 3D-printed vase by Jennifer and I; (G) Klein Bottle by Camila.

(Figure 4.17B). We laser cut 2D template on newsprint and adhered to leather-hard bowl with a wet sponge (Figure 4.17C). We marked each hole position with a probe, removed the template, and drilled holes after drying (Figure 4.17D).

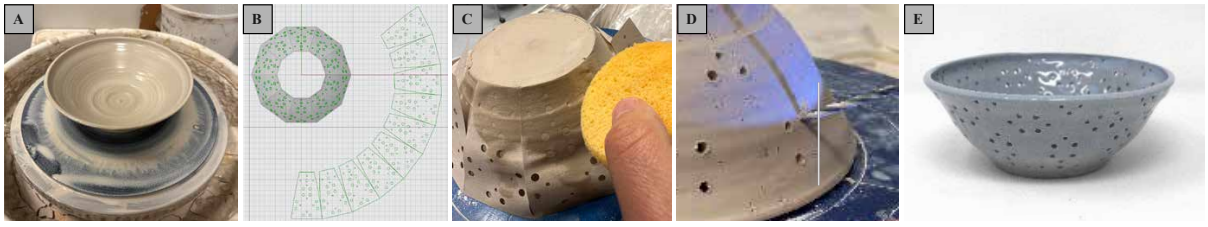


Figure 4.17: Double-curvature colander with two-size drill pattern: (A) Wheel-thrown bowl, (B) global pattern and unrolled 2D template, (C) laser-cut template in newsprint applied using a wet sponge, (D) drilling the surface through stencil-marked locations, (E) the final glazed colander.

This example showcases *CeramWrap*'s versatility in wheel-thrown and subtractive fabrication. It also demonstrates how our workflow can adapt to different tolerance levels in ornamentation. Drilling has higher tolerance than underglaze stenciling because it does not require precise template adherence. We designed a template that conformed to the bowl's global geometry and manually adjusted it to align with its contours. This

reduced fabrication time compared to stenciled examples.

(B) Underglaze on Planar Surfaces of Slab-built Hexagonal Cup: We designed a hexagonal cup in Rhino, with a repeated surface pattern. We unrolled the 3D surfaces, laser cut the template on paper, cut and assembled clay slabs manually. After bisque firing, we measured the cup and updated the digital model. We used *CeramWrap* to generate an ornamentation pattern for the slab-built form. The pattern consists of three triangle elements repeated ten times vertically and six times horizontally (Figure 4.18A–C). The side triangles use a sinusoidal translating function, while the middle one follows a linear function. All three use the same linear function to sequentially scale down the motif. We fabricated the stencil on a vinyl cutter, applied it to the vessel, and brushed red underglaze through it (Figure 4.18D–E).

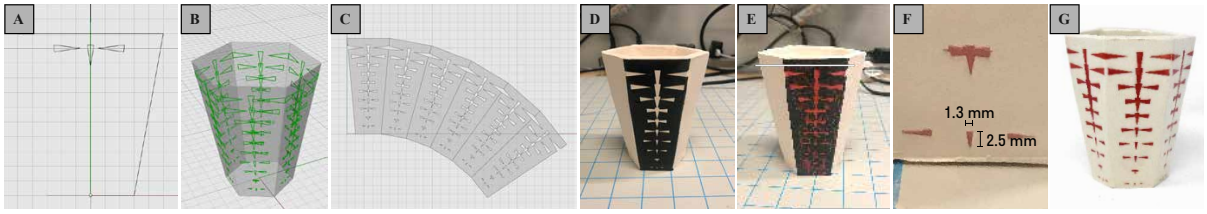


Figure 4.18: Decorating a hexagonal cup with planar surfaces: (A) Triangle base elements, (B) global pattern on the virtual model, (C) unrolled 2D stencil with the pattern, (D) the vinyl-cut stencil is applied on one side of the cup, (E) red underglaze is brushed through the stencil, (F) the precision of details fall in the range of millimeters, (G) the final glazed cup.

This example demonstrates the generation of a vinyl stencil and its application onto a bisque surface of a slab-built planar vessel. It also shows how *CeramWrap* incorporates the precision of computational design and digital fabrication with manual ceramics craft. As shown in Figure 4.18F, this process provided a high level of fidelity within the millimeter range. It further exemplifies the precision of *CeramWrap* by the continuity of the final ornamentation across different surfaces, such as diamond shapes at the top.

(C) Underglaze on Single-curvature Surface of 3D-printed Cylindrical Cup: We apply three-color decoration to a 3D-printed cylinder cup (the workflow stages are shown in Figure 4.8). This example shows the application of *CeramWrap* to a digitally fabricated and manually adjusted vessel with a single curvature. We clay 3D-printed the cup, smoothed the surface at leather-hard phase, and bisque fired it. To account for the shrinkage and deformations, we measured the vessel and updated the 3D model with new measurements as shown in Figure 4.9). We created a nested chevron pattern of varying thicknesses and used it to create three separate vinyl stencils. We aligned the stencils, applied underglazes, and removed stencil sequentially corresponding to each color as seen in Figure 4.19. This approach has the advantage of ensuring complete masking of each color channel during underglaze application but requires a high degree of manual skill and risk damaging the color under the vinyl.

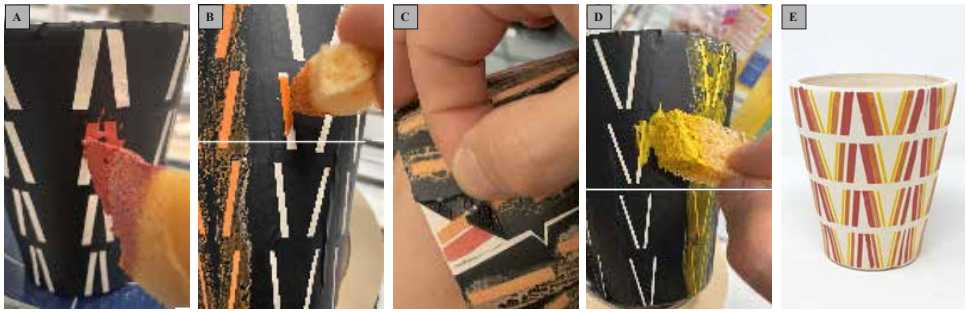


Figure 4.19: Multi-color design using three separate vinyl stencils: (A) Red underglaze through the first vinyl on bisque, (B) aligning the second layer on red layer and applying orange underglaze, (C) removing the orange layer, (D) third layer for yellow underglaze, and (E) the final glazed cup.

This example demonstrates how *CeramWrap* workflow can support multi-colored ornamentation and adapts to support varying forms and degrees of manual skill. *CeramWrap*'s parametric representation of the surface ornamentation and its automated unrolling technique reduces the labor in generating multiple variations of a stencil.

(D) Underglaze on Single and Double-curvature Surfaces of 3D-printed Circular Vase: In this example, we generate a digital model by revolving the red profile curve shown in Figure 4.12A on a circular rail to produce a vase with two double-curvature and three single-curvature surfaces (Figure 4.20). We 3D printed the vase, smoothed the outer wall at leather-hard phase, and bisque fired the piece. We then updated the digital model using the averaged dimensions around the defining points on the profile curve. To create the non-repeating pattern, we extend *CeramWrap*'s pattern-making component to randomize the projection of elements onto the surface and created a grid of 20 horizontal by 5 vertical motifs with groupings of squares and triangles. To account for its curvature, we used a logarithmic function for the vertical translation and an absolute value function for the horizontal scaling parameters to create varying-sized elements around the vase (Figure 4.20A). We approximated the double-curvature region with an icosagon (20-gon), unrolled the 2D templates, vinyl cut unrolled stencils (20 vertical slices for body and 2 continuous horizontal slices for necks), and installed them on the vessel (Figure 4.20B).

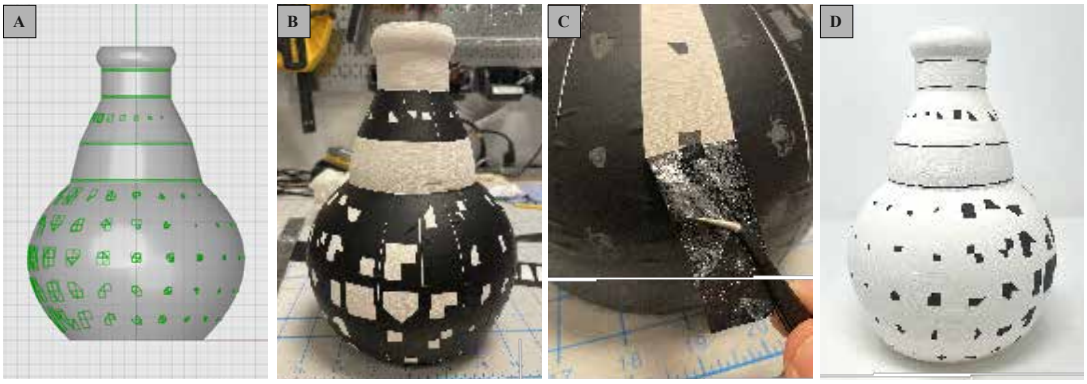


Figure 4.20: Generative geometrical pattern on double-curvature vase: (A) The generative pattern made of squares and triangles, (B) approximating the large double-curvature region with 20 vertical slices, (C) peeling the stencils after black underglaze dries, (D) the final vase.

This example shows how *CeramWrap* supports decorating vessels with complex non-planar geometries through a method for simplifying double-curvature surfaces into single-

curvature stencils. The vertical unrolling approach generates a good approximation of the bulbous surface for our application method; the stencils cover the entire surface and allow for transferring designed pattern onto the surface (Figure 4.20C).

(E) Underglaze on Planar and Single-curvature Surfaces of 3D-printed Octagonal Vase: This example shows ornamentation on an octagonal vase with both planar and single-curvature surfaces (Figure 4.12). This time, we revolved the red profile curve shown in red in Figure 4.12A on an octagonal rail curve. This operation resulted in three planar and two single-curvature surfaces. We used a clay 3D printer to fabricate the vessel and smoothed the outer walls. We bisque fired the vase, measured the physical vessel and updated the digital model. We designed a surface pattern consisting of 135 dots and 6 lines on the digital model (Figure 4.12A) and repeated it on all sides. We vinyl-cut the stencil, manually removed the dots and line geometries, and installed them on the object surface in sections. We added blue underglaze with a sponge through the holes of the stencil (Figure 4.12B) and removed the vinyl (Figure 4.12C).

This example demonstrates how *CeramWrap* can produce continuous and visually cohesive elements across multiple surfaces. We also show the benefits of generating stencils that can be unrolled both horizontally (Figure 4.12B) and vertically to support different decorative layouts and surface types for the same vessel.

(F) Sgraffito on Double-curvature Surfaces of 3D-printed Textured Vessel: This example demonstrates *CeramWrap* can integrate with clay 3D-printed surface textures with sgraffito ornamentation. We use a large double-curvature vase that has three flat regions without surface bumps (Figures 4.21A–B). We repeated the textured layers every 8th layer and modulated the surface texture strength radiating out from the flat surfaces. We sprayed the entire surface with black underglaze (Figure 4.21C) to use

the sgraffito technique to further ornament the surface.

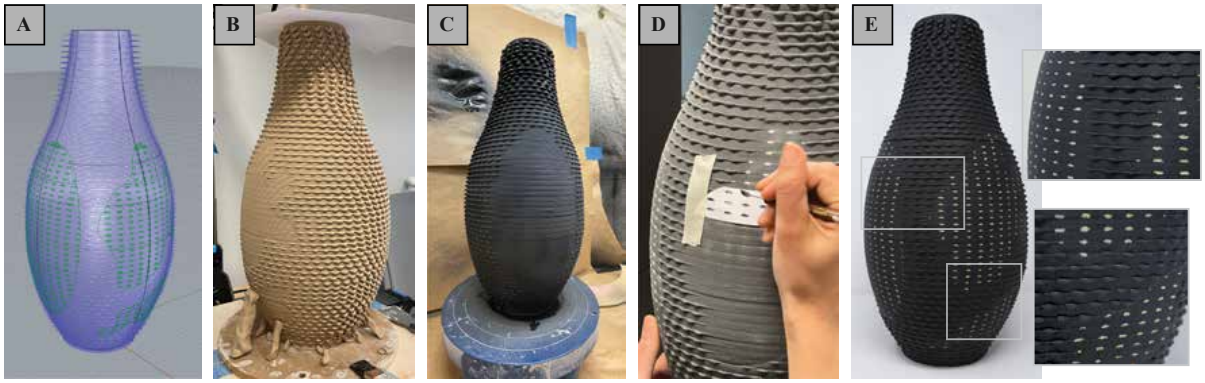


Figure 4.21: Sgraffito on a large 3D-printed vase with displacement-based surface textures: (A) The toolpath (blue) with surface bumps and 3 flat regions; pattern on flat regions (green), (B) 3D-printed vase, (C) spraying black underglaze, (D) applying the laser-cut paper in portions with tape and carving the surface, (E) the final artifact with close-up on surface details.

We discretized the profile curve with an angle tolerance of 11 degrees and therefore converted the smooth profile curve into a polyline with ten segments. We took diameter measurements at vertices and digitally recreated the vessel. For the pattern, we repeated an ellipse-shaped decorative element using linear functions (green elements in Figure 4.21A) on the flat surfaces to align the position and distribution of the 3D-printed bumps. We unrolled the surfaces and laser cut non-adhesive paper templates to avoid removing black underglaze. We sequentially taped each template section to the vessel and used them as guides to scratch off the underglaze with a carving tool (Figure 4.21D).

This example demonstrates how *CeramWrap* can integrate the unique surface ornamentation opportunities of manual and 3D-printed ceramics. We generate a precise template that corresponds with the layers of the 3D-printed vessel while retaining opportunities for manual skill in the sgraffito application.

(G) Carving and Underglaze on Double-curvature Surface of Wheel-thrown Sculpture: This artifact demonstrates the capabilities of this workflow in the hands

of a professional ceramic artist.

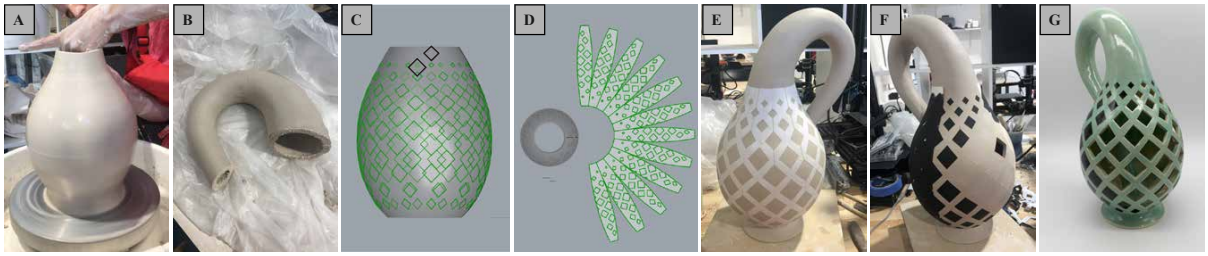


Figure 4.22: Underglaze and carving on one-sided Klein Bottle geometry: (A) Wheel-throwing the body, (B) hand-built ear, (C) digitally recreating the body and repeating the elements in bold, (D) 2D unrolled stencils, (E) installing laser-cut stencils, (F) carving and underglaze painting, (G) the final artifact.

Camila manually fabricated a Klein Bottle, a mathematical non-orientable geometry, in clay by throwing the body on the wheel and hand building the ear (Figure 4.22A-B). After attaching two pieces together and letting it dry, she measured and digitally recreated the artifact by revolving a smooth profile curve on a circular rail. On this 3D model, she repeated diamond elements (shown in bold) on the 3D surface with sinusoidal scaling and linear translation functions (Figure 4.22C). She then unrolled the approximated surface into 2D templates, laser cut stencils, and applied on the body (Figure 4.22D-E). She chose to employ underglaze painting and carving for different parts of the vessel (Figure 4.22F).

This example further demonstrates *CeramWrap*'s adaptability for various decoration methods and complex geometries. The details of this artifact also signifies our method's ability to support expert manual execution.

4.4.3 Parametric or Manual Geometries

Here I present outcomes of *ClayToolkit*'s parametric and sliced geometry support, demonstrating how residents' expert manual skills transformed simpler geometric forms into expressive artifacts.

Radially Symmetric Forms: The most common type of geometry fabricated using *ClayToolkit* was the radially symmetric and parametric forms. They granted an easy entry to clay 3D printing and resonated with traditional ceramic form techniques like wheel throwing. Figures 4.23, 4.27(A,B,C), 4.26(left) show various examples of this geometry type. Combining these forms with surface details demonstrated the precision and repetition affordances of digital clay fabrication. The key parameter that defined



Figure 4.23: Various examples constructed with parametric tool in *ClayToolkit*. (A) Two vessels by Raina. (B) Shot glasses by Avi; made with an early version of *ClayToolkit*. (C) A polygonal vessel by Eun-ha. (D-E) Circular vases in real world location by myself.

these form is the hand-drawn or scaffolded profile curve. This curve is revolved around the +Z unit vector. Objects' spatial parameters change based on the location of this curve. The height is determined from projecting the profile curve onto Z-axis, and radius is interpreted from the its closest point to XY plane. A parameter in *ClayToolkit* determines if the rail curve a circle or a polygon.

Sliced Models: *ClayToolkit*'s slicer tool was instrumental in Eun-ha's workflow as commercially available slicers did not allow adding procedural surface textures. As shown in Figure 4.24, she produced variations of her signature poodle 3D model and glazed them with various colors. The last example of this figure was developed by me as a demonstration of geometry-based texture disable feature of *ClayToolkit* along with a sliced model.

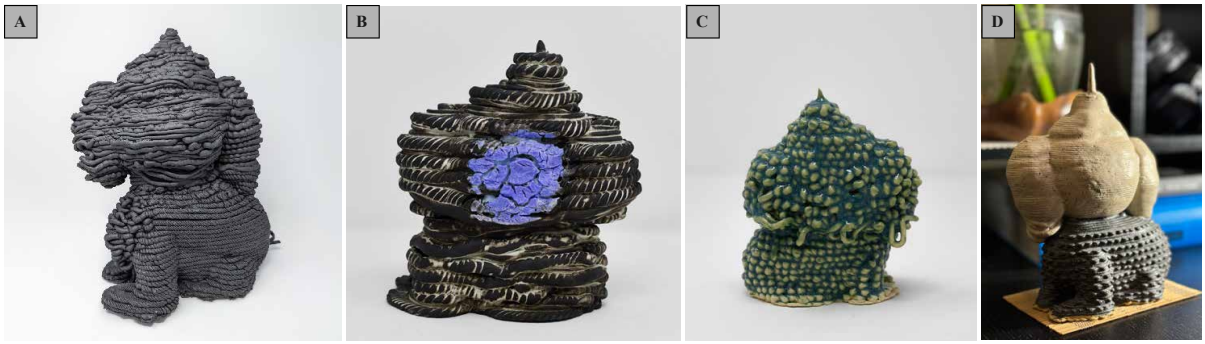


Figure 4.24: A sequence of example Poodles enabled by *ClayToolkit*'s slicer tool: (A-C) Eun-ha's experiments with her signature poodle 3D model. (D) My test print using her 3D model, showcasing the use of geometry-based texture disabling on the head.

Large-scale Fabrication Study: Larger-scale fabrication required different considerations and advanced planning. Figure 4.25 shows the production workflow of Eun-ha's large artifact that was fabricated during the residency, glazed and finished after the completion.

Eun-ha designed the object and printed smaller-scale versions in the lab in preparation for this larger artifact. She used *ClayToolkit* to design the form and to add surface details using an image-mapped displacement-based texturing. The print contained 250 layers with 2 mm layer height.

Multi-wall feature was essential for this print, as the lower half of the spherical object needed to be as strong as possible. In order to avoid straining the print and reduce clay usage, we created two GCode files, one using multiple walls and the other single wall, and manually merged the multi-wall GCode for the bottom half (up to 128th layer) with single-wall GCode for the upper half. Even with this optimization, the clay usage calculator estimated more than 3 tubes required to complete the print.

This print used 3.2 tubes of Soldate clay body with 1 mm/PU softness as measured with a penetrometer, and we manually swapped the tubes at layers 47, 90, and 147. We paused the print before completing to let it firm up. Start to paused point took more

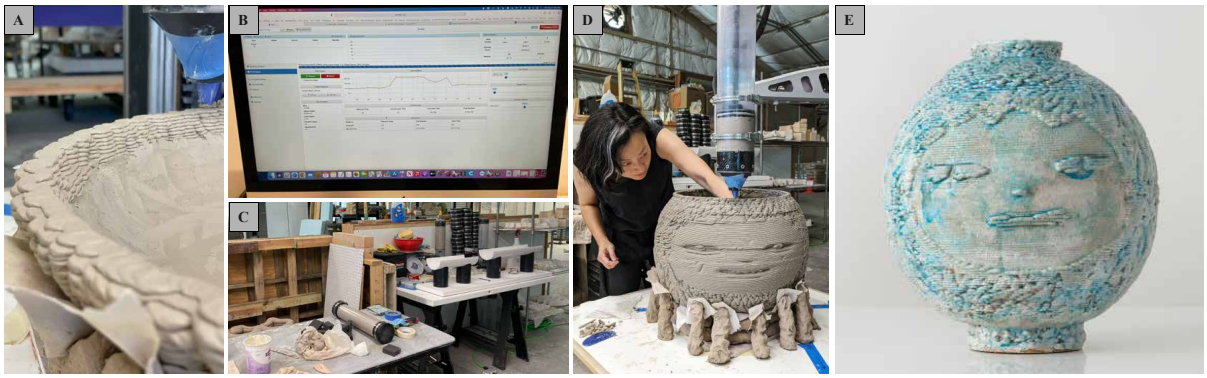


Figure 4.25: An example of large-scale fabrication in *ClayToolkit*, showing Eun-ha's piece printed using PotterBot Scara in the facilities of SB Maker House: (A) A close-up of the ongoing print showing multi-wall feature in *ClayToolkit*. (B) We constantly monitored the print status on Duet interface, adjusted the print speed, recorded and estimated the tube swap timing. (C) Tube swap station positioned next to the printer; used 3.2 large tubes in total. (D) Eun-ha actively attending the print by smoothing the inner surface for stronger layer adhesion. (E) Photo of the glazed and finished piece.

than 9 h. Eun-ha continued the print for another hour once it was slightly more dry.

As described here, going large with clay comes with trade-offs. *ClayToolkit* provided the expressive software to facilitate this workflow as much as possible, and the rest required labor and staying in tune with the ever-changing state of the material. I helped her throughout the file preparation, performed the physical labor of tube swapping, and monitored the print.

4.4.4 Presentation and Professional Contexts

Works described in this chapter have been exhibited in group shows and galleries. We exhibited a selection of the works we produced at *Latentville* exhibition in 2023 in UCSB Elings Hall and in SBCAST. Figure 4.26 shows a selection of these artifacts in SBCAST, along with Sam's CoilCAM pieces. These works reflected traditional formal qualities of ceramics manufacture with additional affordances of digital fabrication technologies made visible in their surface articulations.



Figure 4.26: Various outcomes enables by *ClayToolkit* and *CeramWrap* were exhibited at *Latentville* in 2023 at SBCAST. Smaller objects were developed by Sam using CoilCAM [160].

In addition to *Latentville*, resident artists also chose to include works produced with *ClayToolkit* in their sale portfolios and exhibited in various galleries across the world. Raina included small object she produced during the residency in her tree-house sales, and Avi included some of his smaller works for sale in his mobile store. Residents exhibited some artifacts enabled by *ClayToolkit* in global exhibitions (Figure 4.27). Raina glazed many of her 3D printed pieces following the residency and exhibited them at the LaiSun Keane Gallery in Boston in May 2024. Eun-ha showed Obsidian poodle (Figure 4.24A) at Moosey Gallery, London, and various *ClayToolkit* works in a solo show at HB381 Gallery, NY.

These works demonstrate the expressive potential of integrating material behavior in digital contexts. In this chapter, the human decisions guide most of the material behavior. The results section of Chapter 5 adds to this by incorporating generative algorithms for constructing life-like behaviors. *ClayToolkit* and *CeramWrap* was essential in building the tools and techniques required to work with more complex geometries and hybrid emergent workflows.

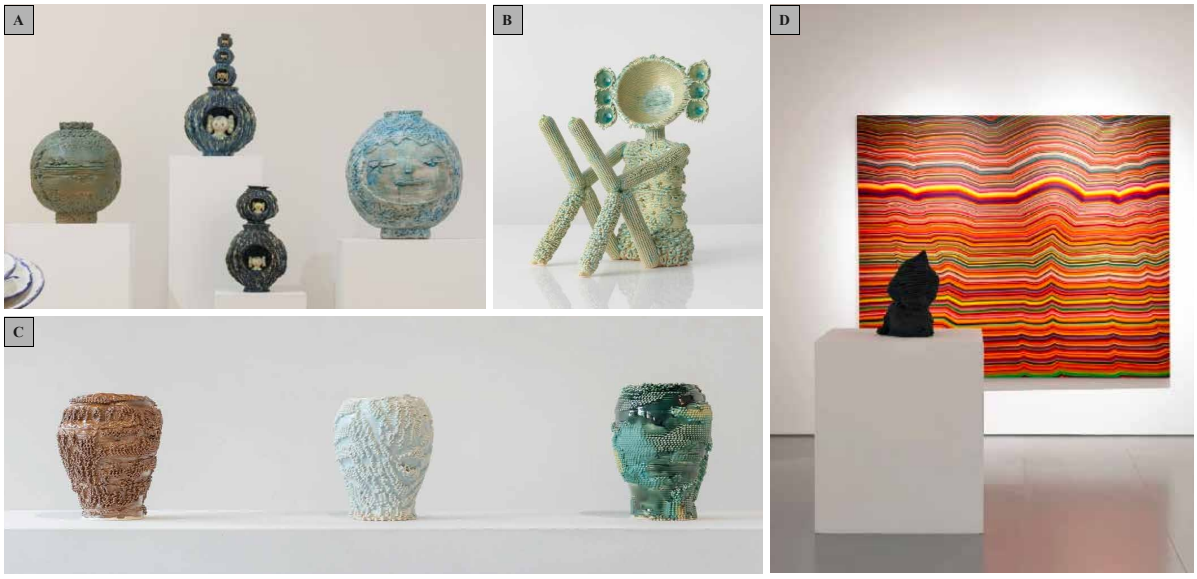


Figure 4.27: Residents exhibited pieces made by *ClayToolkit* following the residencies: (A-B) Eun-ha exhibited a new assembled artifact, the large-scale artifact in Figure 4.25, and another spherical work in her 2025 solo show *Alters* at HB381 Gallery, NY; only spherical works in (A) have been made by *ClayToolkit*. (C) Raina glazed her image-textured series and exhibited in the 2024 group show *Calibration* in LaiSun Keane Gallery, Boston. (D) Eun-ha exhibited her sliced and textured Obsidian poodle in 2023 group show *Well Cold* at Moosey Gallery, London.

4.5 Discussion

This discussion reflects on the opportunities, challenges, and limitations of software developed for clay fabrication with particular insights gained from *ClayToolkit* and *CeramWrap*, but also from the residency at large.

4.5.1 Practice-driven Software Development for Material Production

Tools shown in Figure 4.2 collectively revealed insights into software *ideation, implementation, and debugging* for targeting material constrains and physical practices. Below, I present a categorization of these software development activities with example interactions with resident artists and other researchers who developed software along-

side with me. Although these activities are presented linearly, they often overlapped or occurred iteratively.

Ideation

Software development often starts with ideation, and new software ideas in this work came through direct requests from artists, in response to acute breakdowns, and through semi-structured observation and dialogue.

Much of the software ideas were based on resident artists' direct requests. They expressed desire to use specific design elements or to explore new aesthetics that pushes their creative agency in the face of software tools. For example, after experimenting with displacement-based surface textures on procedural geometry, Eun-ha wanted to apply such textures on her signature poodle 3D model (Figure 4.5). This motivation led to the creation of *ClayToolkit*'s *slicer tool* explained in Section 4.2. Other ideas were developed in assistance to challenges of performing specific tasks. We conducted ad-hoc group brainstorming sessions to pinpoint digital, mechanical, or material solutions to the problem. For example, Pilar and Avi experienced under-extrusion on PotterBot early in the residency, leading me to add the software feature explained in Section 4.2.2 that calculates the correct extrusion rate for different nozzle sizes.

Some ideas for software features were identified through observation and physical making. Ideas emerged during group meetings, digital design, and fabrication. Unlike plastic 3D printing, clay prints require continuous monitoring to tune machine parameters and add manual supports to prevent collapse. Printing became a site of collaborative ideation as the team observed failures and proposed software approaches to exploit or compensate for material behavior.

The residency allowed researchers to gradually develop a tacit understanding of the residents' software-fabrication workflow through observation and collaboration. This

understanding prompted us to develop new techniques without explicit direction. For example, I added a boolean difference feature in *ClayToolkit* for disabling surface textures that intersect with a pre-defined geometry (Figure 4.5B), motivated by my observations of the residents' efforts to configure image maps in direct-manipulation software to control textures locally.

Implementation, Fabrication, and Feedback

The development workflows were characterized by tight iteration cycles alternating between writing code and performing fabrication tests. We quickly found that clay-specific software implementation must be regularly interspersed with fabrication because software cannot adequately simulate the behavior of clay bodies. For example, we found that surface textures manifest differently depending on the clay composition, extrusion rate, nozzle diameter, and layer height. These differences impacted aesthetics and print viability, making it critical to determine acceptable design parameters. Residents communicated their technical expertise through manually and digitally fabricated objects by referencing specific physical features and material qualities. Similarly, researchers used physical objects to compare the effects of different software implementations and solicit feedback from residents. We found that this bidirectional information flow rarely occurred when working exclusively in the digital domain.

Researchers started software implementation by quickly writing code for viable toolpaths. Material testing was usually needed before digital work. For example, while developing the *CeramWrap* workflow, we explored screen-printed glazes, and when failed, we pivoted to testing decoration with cut stencils. We then developed a computational design workflow compatible with subtractive digital fabrication. As our software tools matured and our familiarity with clay 3D printing increased, we reduced iteration frequency between coding and fabrication.

Our development accelerated as residencies progressed because researchers could reuse functionality from another tool within new tools. Clay 3D printing has a core set of operations that software developers can use in different forms across software technologies. Examples include defining toolpath structure, adding texture, specifying printer settings, and tuning print behavior. Reusing code across multiple tools substantially accelerated our development in 2023 when many low-level utilities were well-tested across multiple machines and materials. For instance, *CoilCAM* and *SketchPath* used *ClayToolkit* 's GCode generation utilities due to its support for versatile extrusion calculations. When Raina and Eun-ha wanted to apply displacement-based surface textures to 3D models, I reused codes from Jennifer's computational fabrication class examples and *ClayToolkit*'s surface textures module to develop its *slicer tool*.

Debugging

The residency format led to different emphases in debugging than typical systems research. Because residents were using the software daily, software errors that led to print failure or extrusion errors became our highest priority. When researchers could not fully resolve an error, residents found ways to salvage or repurpose what would otherwise be a failure. For example, while experimenting with image mapping on cylindrical forms, Eun-ha encountered an issue where the printer printed the concentric circles of the base separately, resulting in a non-functional base. Instead of debugging the code's extrusion rate or line separation, Eun-ha discarded the faulty bases and later assembled individual cylindrical pieces into a larger sculpture. I later fixed this bug in *ClayToolkit* by making sure the separation of concentric lines functions well with various extrusion widths and nozzle sizes (Section 4.2.1).

In other cases of software failure, residents resorted to manual repair rather than relying on a software-based solution. While printing a parametric variation from *ParaLight*

software, the central portion of the vessel collapsed. Pilar paused, manually repaired, and resumed the print, but in doing so, shifted the geometry. This misalignment with toolpath and ongoing print required manual guidance of the extruded coil for several layers to ensure they overlap. Factoring in printing time and clay preparation, Pilar's repair was more efficient than adjusting parameters and starting a new print.

Digital fabrication systems have multiple points of failure, including errors in geometry design and toolpath specification, mechanical breakdowns and vibrations, and inconsistencies in material qualities. Research team often reasoned about failures based on their expertise; while residents examined material factors like clay type, moisture content, and plasticity, researchers considered software issues that may have led to undesirable tool trajectories or incorrect material deposition. For instance, Eun-ha printed unexpected braided coil-like textures while printing variations of a form with surface textures. I assumed this was due to a software bug, but further investigation revealed it was caused by how the clay behaved in response to Eun-ha's software parameters. Material-based reasoning often revealed more effective debugging outcomes than focusing solely on software implementation.

4.5.2 How Fabrication Labor Mediates Software Production, and When Software Cannot Address Clay 3D Printers' Limitations

Qualitative analysis of the data collected through the residencies revealed key areas of interconnected relationships between the software production and fabrication labor, as well as the limits of software addressing the physical issues with clay 3D printing.

During the residencies, artists had to adapt to new production workflows and academic researchers had to shift their understanding of digital fabrication workflows in

response to the unique demands of clay 3D printing. In this environment, software development was shaped by the *physical* and *design labor* requirements of moving between manual clay fabrication and 3D printing.

Physical Labor: Clay 3D printing, unlike residents' initial expectations, demands significant physical labor. Loading the large PotterBot tubes with 18 pounds of clay is time-consuming and physically taxing (Figure 4.6). Raina described the process, emphasizing its craft aspect and distinguishing it from purely "click-and-print" 3D printing. While everyone engaged in clay preparation, residents produced a higher volume of work, requiring more physical labor for tube loading. This labor shaped their attitudes toward printing. Raina noted that the demanding clay tube loading process made each print more precious. Additionally, Raina found the production pace could not match her manual skills, as she could throw 10 cups in an hour on the wheel but she could not do so with printing. Existing software tools offered little support for planning clay tube switches. Pilar mentioned that the model with four layers ran out of clay quickly, leaving a short post, and she had to plan for tube switches and start again. When clay usage was a factor, residents faced new challenges in planning and iteration.

Watching residents' daily physical struggles motivated software solutions to ease some of the strain. I implemented a clay usage calculator in *ClayToolkit* described in Section 4.2.2 that calculates the tube percentage needed to print a digital form. This ability influenced Raina's aesthetic decisions. Similarly, initial software versions before residencies lacked functionality for printing watertight bases. Residents manually added the necessary bases for functional ceramics and prompted to develop a printable base with a repeating spiral pattern. Avi identified flaws in this design, showing how the lack of overlap would lead to cracking. He advised interspersing linear and spiral toolpaths in a 3-4 layered structure, which was implemented in software going forward.

These patterns show how clay 3D printing subverted reduced manual labor expectations through software automation. Instead, software development was bounded by the physical labor of preparing and managing printing material. The research team's exposure to this labor and its strain incentivized software development to streamline form production and planning.

Digital Design Labor: Residents regularly reported their experiences shifting their design process from a physical to a digital space. Pilar described the loss of "immediacy" and embodied collaboration with material, and contrasted this with an unintuitive, abstract "XYZ" design space. Raina likened early software use to "traveling through a foreign country", which motivated embodied tools such as *SketchPath* [161]. Both Raina and Eun-ha used *SketchPath* when they needed a break from wrangling Grasshopper. In addition, although not as embodied as *SketchPath*, *ClayToolkit*'s parametric design approach over generative methods was an attempt to ease complicated digital design labor for the residents.

Residents embraced learning new digital workflows but benefited from working with technologies aligned with their physical design experience. Researchers' dialogue with residents about the laborious shifts from physical to digital and symbolic modes of manipulation inspired them to create new technologies blending embodied production. Developing such tools involved multiple residency iterations.

Unaddressable Limitations of Printing in Software: The residencies revealed key areas in which software-based solutions cannot address challenges in clay 3D printing due to specifics of material properties.

Clay 3D printing, like all forms of digital fabrication, exhibits material and workflow limitations. Material limitations comprise issues stemming from the clay state and be-

havior, like plasticity and moisture. While the plasticity of clay is a design affordance, it also impacts a print's success by affecting its structural stability and constraining its geometry. In digital fabrication domains with more uniform and stable materials, digital simulation can provide a partial means to predict artifact performance in relation to material properties. Such predictive modeling tools do not exist for clay 3D printing, nor did we attempt to engineer them because clay hardness and viscosity are constantly in flux. Raina prioritized reliability by adopting cylinders to avoid failures.

Workflow limitations determine how the aesthetics, functionality, and production effort for clay 3D-printed objects compare to the manually fabricated objects. The robustness of stacked layers of a 3D-printed piece is largely a function of the compression and alignment across layers. Insufficient overlap will result in cracking and separation during firing. In response to layer separation on some of his cups, Avi noted that layered lines can become a "weak point" compared to thrown work. Pilar observed that some motions achievable by hand, such as a printer that could "spiral its coils", can confer "strength" and durability. Like many artists who clay 3D print, we used a simple solution to address layer delamination by decreasing layer height. This method increases the compression of layers but creates undesirable surface artifacts and substantially reduces textural expressiveness.

While software can help support new design workflows and provide alternative fabrication strategies, it cannot compensate for fundamental limitations in mechanical functionality or reliably predict complex material behavior. These limitations indicate the opportunity to conduct future research in clay 3D printing hardware design and material development; however, they also provide an important reminder of the significant gap between skilled manual manipulation and digital machining.

Manual Solutions to Digital Problems: In many cases, the residents solved printing problems efficiently through manual intervention rather than waiting for a software-based solution. For instance, Pilar was not fully satisfied with the results of *Para-Light* [49]. She post-processed them heavily after the print, modifying the forms' curves and smoothing the surface to remove layer lines. She often added coils and used hand-building tools to refine the surface. Avi centered 3D-printed cups on the wheel and reshaped rims to the correct geometry for drinking, noting that simply "throwing a rim" brought the 3D-printed work "to life".

Eun-ha frequently printed multiple components to manually assemble. Assembly allowed her to produce works larger than the size constraints of the printer, create more extreme geometry than would survive in a single print, and streamline her tube usage. After a mid-print collapse with the Obsidian poodle (Figure 4.24A), she attached two parts and added loose coils at the seam, preferring this over a longer digital setup.

Residents also repurposed software in unexpected ways to address printing limitations. Frustrated with prints with overhangs collapsing but interested in exploring alternative geometries, Raina used image mapping on extruded cube towers and later manually separated the sides to create flat reliefs with 3D-printed textures (Figure 4.15F). Seeking non-linear changes in layer height, Eun-ha used the baby-stepping feature of the PotterBot control interface to move the print head in the Z-axis during a print; "very easy to do manually" but difficult to encode in software.

The residents' manual solutions to clay 3D printing limitations demonstrated how 3D printing could complement skilled manual production more efficiently than waiting for software-based solutions. Seeing the residents implement manual solutions allowed the research team to focus on developing software that maximized the unique affordances of the printers and better understand the portions of the workflow that professional practitioners could perform more efficiently through manual labor.

4.5.3 Limitations

As discussed throughout this chapter, clay fabrication is a labor intensive domain of digital fabrication that builds on top of established craft practices. Working with clay using PotterBot printers presents constraints such as continuous extrusion and relatively large nozzle dimensions. The material behavior both pose limitations and present opportunities in these constraints. For example, smaller nozzle sizes like 1 mm exist but making the clay soft enough to extrude from this opening makes building stacks of layers challenging. However, continued workability of clay and skilled manual modifications can overcome some of these limitations (e.g. removing the bridging elements in continuous extrusion manually).

This chapter primarily investigates manual-computational methods where digital clay fabrication takes place. The forms I use here are either manually-made or procedurally-generated, and does not reflect the life-like formal qualities discussed in Chapter 3. For example, *CeramWrap* workflow is limited to radially-symmetrical vessels. This reduces the modeling process to just a few steps, and radial symmetry is extremely prevalent in ceramics (e.g. wheel throwing). Limiting the application space to these forms allowed for examining specific challenges related to surface ornamentation and material agencies.

CeramWrap's radial-symmetry condition creates a discrepancy with the evolving physical material state that displays physical distortions and non-uniform shrinkage. Addressing this discrepancy through a non-linear computational workflow complements the dynamic nature of ceramics craft more meaningfully than attempting to reproduce a dynamic physical object in a digital model. Our workflow is tailored to decorative elements that can be organized in a rectangular grid or arranged in a grid-free manner. This produces various effects aligned with manual ceramics patterning. At the moment, *CeramWrap* does not support raster designs or non-rectangular tessellations but the po-

tential for using these design with generative frameworks is a fruitful direction for future work.

Portions of *ClayToolkit* overlap with the existing clay 3D printing design tool Potterware [223] and general-purpose slicers. Restructuring these functionalities as modular components within a toolkit rather than using proprietary software allowed creating different toolpath processing workflows on the fly and rapidly adapting to artists' design and fabrication needs. Furthermore, *ClayToolkit* significantly reduced the development labor of successive software tools by providing utilities for toolpath analysis and preparation. For example, the effective-extrusion experiment proved useful across all systems developed during the residencies. There is a chance that the calculated rates might not produce accurate results for every clay body and moisture content. However, it did not produce excessively faulty extrusion amounts during three years of extensive use by more than ten people producing hundreds of prints.

In this chapter, I primarily aim to support professional artists' creative agency with *ClayToolkit* and *CeramWrap*. These artists' material know-how informed many aspects of these systems that later feed into the projects I describe in Chapter 5, in which I cover the space of generative form-making in clay fabrication.

To summarize, this chapter lays out the realities of interfacing software development and physical production using clay by discussing the development of two computational systems, *ClayToolkit* and *CeramWrap*. These systems were developed in the context of two artist residencies and have helped produce many artistic outcomes. Gaining the tacit knowledge about the materials is an arduous task that requires years of focused training and making. Working in collaboration with professional artists gave me the insights needed for developing software for craft applications. Many features have been developed

in response to supporting artists design intent and creative agency, while learning and developing computational techniques to work with the material agency of clay. This chapter primarily relied on deterministic digital geometries to study how material and human agencies can contribute into the emergent form development. In the next chapter, I describe methodologies into combining computational agencies described in Chapter 3 and material practices described in Chapter 4 to synthesize what I call distributed creative agencies between human, computational, and material actors into tangible projects.

Chapter 5

Distributing Agency between Human, Computational, and Material Actors

The previous two chapters introduced two complementary perspectives on emergent expression. Chapter 3 examined computational agency by demonstrating how artificial life algorithms hold emergent design possibilities and how human maker can orchestrate and control these emergent outcomes in a virtual context. Chapter 4 investigated material agency and showed how the properties and behavior of clay can redirect deterministic digital intentions and how professional craftspeople can help integrate computational steps in adaptable physical production workflows. This chapter primarily relied on parametric digital design methods in order to isolate material operations that reveals, as Tihanyi writes, the "tug-of-war between the creative agencies of the material and the artist" [20].

Integrating computational and material processes exposes practical limitations (e.g. structural fluidity, non-uniform shrinkage, and machine constraints), while also offering new opportunities for haptic feedback and surface articulation. Studying these com-

plex interactions provides insights into how creative design decisions can be distributed among human, computational, and material actors. In this chapter, I investigate this hybrid expressive space emerging through *distributed creative agencies* by presenting two case studies: *Morphogenic Clay Vessels* and *Distributed Agency Trails*. These projects synthesize methods from the previous chapters and discuss the implications of combining the material agency with emergent computational processes. It iterates the challenges and opportunities of using clay in fabricating computationally generated, artificial life forms.

This chapter integrates the partial responses to my research question from the previous chapters. It primarily addresses RQ1 by integrating morphogenetic making described in the material domain with morphogenesis algorithms described in the generative art domain. This synthesis identifies specific computational and material procedures that enable an emergent expression in hybrid morphogenetic design spaces. It addresses RQ2 by integrating methods described in the previous two chapters that investigated creative agencies between human-computational and human-material. It documents how creative agency is redistributed between human and nonhuman actors that negotiates orchestrating the emergent results into collaborative artifacts. And lastly, it uses the previous chapters' methods to address RQ3 to introduce new tools and workflows that support artists collaborating with computational and material agencies simultaneously.

5.1 Methodology

In this chapter, I use a practice-based research methodology [208] through implementation and artistic use of generative systems in conjunction with physical fabrication. These computational systems display emergent aesthetics and computational agency through life-like behaviors, and their physical instantiations through digital fabrication

reveal additional generative qualities coming from the material properties and behaviors. Combining these two spaces prototypes hybrid morphogenetic making through system implementation, control strategies, and producing artifacts and exhibitions that form the basis for contribution to media knowledge.

This chapter discusses two projects: The first, *Morphogenic Clay Vessels*, adapts a digital morphogenesis algorithm for clay 3D printing to produce undulating forms resembling coral reefs. I developed this project in 2022, produced initial artifacts with varying success levels. This formative experimentation of working with clay fabrication revealed challenges and opportunities that prompted an in-depth exploration of its tacit material knowledge and established workflows. These explorations have been discussed in Chapter 4.

Throughout the residencies, *ClayToolkit* (Section 4.2) acted as a repository of the research lab's growing knowledge about clay's material and workflow knowledge and allowed embedding part of this fabrication experience in software abstractions. I used this knowledge and software primitives to implement a second prototype project in 2025, *Distributed Agency Trails*. It uses a custom generative algorithm to treat 3D printer nozzle as a computational agent with its behavioral rules. The resulting generative computational forms, when clay 3D printed, revealed additional emergent qualities in their physical realizations.

These projects have been exhibited in two exhibitions:

- SyMADES, *Santa Barbara, CA*, 2022
- DeepCuts, *Santa Barbara, CA*, 2025

5.2 Case Study: *Morphogenic Clay Vessels*

This prototype details my first exploration of using generative computational systems with clay fabrication. It uses a digital morphogenesis algorithm (i.e. differential growth [224]) applied to the construction of a clay vessel, resulting in geometries that resemble coral reefs to explore the aesthetic possibilities of emergent computation and material actors. This formative project made visible the challenges and particularities of clay 3D printing and enabled a deeper investigation in manual practices and techniques.

I operationalized the constraints of clay 3D printing with PotterBots outlined in Chapter 4 (i.e. clay plasticity, continuous extrusion, self-support requirement) for constructing a CAM-based generative algorithm. In this project, I use the temporal unfolding of the algorithm to construct vessel's Z-axis to turn its time-based evolution into the geometry configuration. This approach builds on one of the digital approaches for 3D form construction outlined in the *Edge of Chaos* (Section 3.4) and uses it in conjunction with CAM-based fabrication methods outlined in *ClayToolkit* (Section 4.2).

5.2.1 System Design & Implementation

Differential growth is a well-known generative algorithm that iteratively grows a given starting shape similar to biological growth mechanisms [225]. This project uses custom Python components inside Grasshopper/Rhino environment that specifies initial conditions and growth parameters. This environmental construction provides a high degree of computational autonomy, while Rhino's external geometry-based control options provides an opportunity for fine-grained control on the emergent results. Keeping the fabrication process in mind, the digital environment is designed to stack 2D differential growth simulations as layers with small deviations from the previous shape. There could be a large variation between the first and last layers.

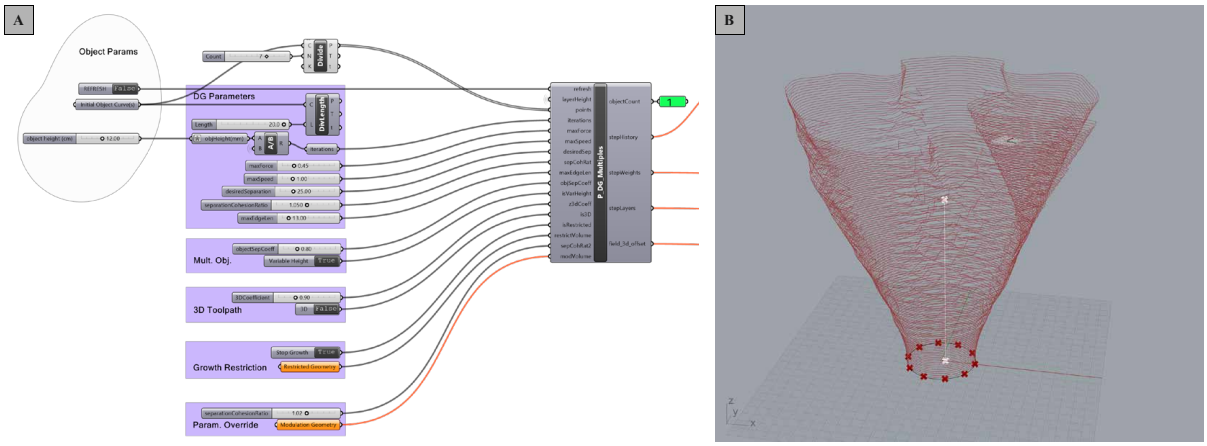


Figure 5.1: The main interface of *Morphogenic Clay Vessels*: (A) Grasshopper interface with the custom differential growth generator on the right; fabrication utilities are omitted for brevity. (B) Visualized toolpath with starting geometry displayed at the bottom; initial nodes are marked with cross shapes; center of mass projection is marked with white cross.

For simulating the 2D differential growth, I adapt an approach called 'differential line growth' that requires `nodes` and connecting `edges` interact primarily with `cohesion` and `separation` forces. If the cohesion force is larger than the separation, rather than displaying growth characteristics, objects shrink. These forces are similar to the flocking forces described in Section 3.2.1. I later expanded this algorithm to include object types that would allow growing multiple shapes grow together while avoiding collisions with one another. These building blocks can demonstrate incredible variety when they coupled with different initial geometries and parameter values.

In order to simulate the 2D growth, the algorithm starts with an initial distribution of nodes and edges that makes up the starting geometry. In *Morphogenic Clay Vessels*, I sample the nodes from `inputCurve(s)` based on the `pointDistance` parameter. Most of my experiments used a circular distribution for its simplicity and contrast between the undulated grown layers at the top of the vessel. Other starting distributions were useful for specific applications.

In each step of the update loop, the nodes interact with other nodes through the

separation and cohesion forces. The movement speed and individual forces applied on each node is capped using the `maxForce` and `maxSpeed` parameters. When an edge grows longer than a parameterized `lengthThreshold` value, a new node is added in the mid point of this edge. A combination of these forces pushes nodes away while maintaining a minimum distance with each other. The addition of new nodes increases detail and introduces spatial morphing, buckling and growth to the original shape.

Mapping Evolutionary Time to Space

In order to construct a 3D geometry suitable for fabrication using 2D differential growth, I stacked the result of simulation steps in Z-axis. The distance between these layers in Z-axis is determined by the `layerHeight` variable in the printer configuration. The system accepts an input for `objectHeight` in millimeters, and determines the number of simulation steps required to construct this height based on the `layerHeight` and a modulo with `stepsPerLayer`. When this modulo operator > 1, each layer demonstrates greater variation and results in a larger overall expansion. I primarily kept this parameter at default value of 1 to avoid higher chance of collapse during the fabrication process.

As differential growth evolves, nodes in the original distribution are carried out to the top, and each time step adds new nodes into the curve. Another time representation in *Morphogenic Clay Vessels* is to visualize vertex 'age' using displacement-based surface textures (Figure 5.11B). This routine hints at a stronger connection between the virtual and the material forms.

Non-planar Layers

In essence, the core growth algorithm in *Morphogenic Clay Vessels* does not differentiate between 2D or 3D growth. If allowed, a planar starting layer can also grow in Z-dimension. This allows for layers deviating from the XY-plane and creates movement

in Z-axis. *Morphogenic Clay Vessels* uses a `3dCoefficient` parameter $\in [0, 1]$ to control this motion. A value of 0 makes the layers perfectly planar and 1 makes the algorithm operate in 3D. Values closer to 0 produce a subtle wobble in Z-axis.

However, the mechanical limitations of 3D printer nozzle prevent utilizing this algorithm with its full degrees of freedom. I primarily tested this approach with a very small value of the 3D coefficient. Even with values $<< 0.1$, the initial planar layer starts growing in Z-axis, and stacking these layers on top of each other introduces greater Z-variations towards the top of the vessel. Large variations limit the fabricability of the toolpath as explained in Section 5.2.2.

Growing Multiple Objects

Another modification to the original algorithm was the ability to grow multiple systems in the same update loop. This is to grow objects with similar qualities that fits into each other like puzzle pieces.

As the input curves passed into the algorithm, the nodes of the distinct curves are assigned incremental an `objectIndex` in the constructor. Even though there is no official collision detection between different objects, *Morphogenic Clay Vessels* utilizes the separation force calculations to avoid collisions. As the main growth loop iterates over the nodes, if two nodes have different object indices, the separation forces gets multiplied by an input parameter `objectSeparationCoefficient`. While this value is greater than 1, the nodes of distinct object pushes away and creates a separation between the objects.

My initial tests with these multiple geometries demonstrated promising results, however the default system produced objects at the same height. This is desirable in some cases, but modulating the height also showed promise. I added a `variableHeight` parameter to stop the growth of certain object indices prematurely using two approaches: (1) passing the height values in millimeters as an array, and (2) using a mapping function

to determine the heights mathematically (e.g. $h_i = h_g * (-|i/(N+2)|^{0.75} + 1)$; where i is object index, N is total number of objects, h_i is the height of i-th object, and h_g is the global height value). Objects produced with this mapping function are presented in Figure 5.10.

Controlling the Growth Parameters

Even though the differential growth produces emergent results, *Morphogenic Clay Vessels* employs various strategies to control the growth parameters locally or globally to introduce variations in the outcomes.

One strategy is blocking the growth of specific nodes that falls within a geometry. This geometry is created using the Rhino viewport and passed into the custom Python component as a parameter. When it is enabled, the Python component stops applying the cohesion/separation force calculations to the nodes that intersect with geometry, and produces a series of static nodes. This geometry can be a solid shape or a mesh. Example objects are shown in Figure 5.11A.

Similarly, the same intersecting geometry approach is used to modify growth parameters. Rather than stopping the growth, this input modifies the growth parameter values of the nodes that fall within the geometry. While any parameter can be defined for this modification, I primarily used `separationCohesionRatio` to be able to selectively grow or shrink the object at custom locations. This produced local pinching or spreading of the nodes within the volume. Example objects are shown in Figure 5.11C.

5.2.2 Physical Production Workflow

Morphogenic Clay Vessels contains fabrication-related components of *ClayToolkit* (Section 4.2) that enable specifying printer configurations, creating displacement-based

surface textures on the vessels, and producing the machine instructions as GCode. These components does not offer computational autonomy, yet they help material's agency to display emergent properties in the final outcome. The combination of parametric and generative design methods produced prominent results shown in Section 5.4.

Toolpath and GCode Generation

The script described above returns a Grasshopper tree containing the `layerIndex` and an ordered array of `nodeCoordinates`. In order to create a continuous toolpath, the Z value of each node is incremented by $\Delta z = \text{layerHeight}/N$, where N is the total number of points in the layer. To avoid generating an over-extruded seam, the last point of lower layer is removed because, after spiraling, it duplicates the first point of the next layer.

Once the toolpath is spiralized, GCode generator (described in Section 4.2) reads the printer specifications and the toolpath to convert digital coordinates to machine readable instructions that contain material extrusion, machine movement and speed values. I primarily used constant extrusion (`E` value) for this project, however I applied a variable speed calculation (`F` value). In the GCode creator, I use the spiralized coordinates to calculate the angles between two line segments and apply these values to to a linear mapping function to slow down the machine movement at sharp turns ($f : [0, \pi]$ rad $\rightarrow$ $[150, 3000]$ mm/min). I implemented this routine after observing sudden mechanical motion that shakes the printer bed and displaces the ongoing clay print. Reducing the speed at sharper turns helped reduce this sudden motion.

Mechanical Limitations of Nonplanar Printing

PotterBot with three degrees of freedom assume the nozzle is primarily moving on the XY plane. It is possible to supply varying `Z` values in the GCode file to move nozzle



Figure 5.2: Nonplanar printing with PotterBot is limited with nozzle geometry: (A) Nonplanar printing experiments of three objects (top) wet and (bottom) bisque-fired versions; surface is heavily disrupted with large variations in Z. (B) printing a non-planar differential growth vessel with minor variations in Z axis. (C) Glazed ceramic.

vertically to print non-planer geometry. I used this approach with 3D version of the differential growth. However, the main limitation for pushing this direction forward was the geometry of the physical nozzle. In my experiments, the angle of vertical motion before the nozzle hits printed clay was less than 10 degrees. In most cases, increasing the `3dCoefficient` of the generative system past 0.05 introduced too much nozzle collision and destroyed the print. I partially addressed this issue by creating DIY nozzle with a longer tip, but there were so much clay accumulation inside this small tip that I had to give up on this approach.

Clay Fabrication Workflow

I have collected data about 37 clay 3D prints enabled by *Morphogenic Clay Vessels* between January and July 2022. Here I present this dataset, noting how material behavior and agency was observed throughout the ceramic pipeline (see Section 2.3.2 for summary) with respect to different clay bodies, fabrication techniques, and kiln environments.

I loaded 6 tubes for PotterBot Super 10 by hands and a lever-based hand extruder

($\sim$3000 cm^3/180 in^3 clay per tube):

Clay Body	# of Prints	Fire Temp.	Added Water	Notes
Obsidian	8	Cone 5	230 mL	black, smooth porcelain clay
Sandstone	9	Cone 06	230 mL	low-fire stoneware
SBFG	14	Cone 5	460 mL	groggy stoneware
Bone White	2	Cone 5	460 mL	white, smooth stoneware
2 x FSB	4	Cone 5	none	very groggy sculptural clay

Table 5.1: A table of clay bodies used for fabricating *Morphogenic Clay Vessels*.

The layer height directly affects the form due to stacking individual timesteps of growth in the Z-axis. It also significantly impacts artifact strength and surface qualities. During *Morphogenic Clay Vessels* fabrication, I experimented with layer heights in the range 0.5 mm to 4 mm, with 1.5 mm being most common. Higher layer heights produced larger vessels faster, while lower ones ensured sufficient compression between layers to prevent further failure. However, a layer height of 0.5 mm resulted in the disappearance of canonical 3D printed lines and a stone-like surface due to clay cracking.

I used nozzle diameters in the range 1 mm to 8 mm throughout the prints. The 8 mm nozzle was exclusively used with FSB due to its grog content, while the 1 mm nozzle was only used with Bone White due to its softness but insufficient layer area for compression and larger objects (Table 5.1). The 3 mm nozzle produced the most results and supported the weight of medium-sized objects. While experimenting with nonplanar toolpaths, I attached a thinner and longer 2.8 mm LUTUM nozzle attachment through the 4 mm PotterBot nozzle. This extra nozzle comfortably fit, but clay accumulation in the connection piece rendered this approach impractical for a reliable use. Future work will experiment with creating custom nozzles designed for nonplanar printing.

After printing, I covered the objects and let them dry on the printed bats for varying times. Depending on size and wall thickness, drying took 1 d to 7 d in the lab at ambient temperature (22 °C) and humidity (31 %). For some objects, I attached bases while they

were leather-hard. I also fixed some objects with layer separation by adding moisture and manually compressing the detached layers that happened due to insufficient layer compression while printing due to large layer heights. Most of these prints did not succeed at the end because of my rudimentary approach to manually fixing clay objects.

Once the objects were dry enough, I fired most of them to final temperatures around Cone 5 without glazing. Only a few objects were bisque fired at Cone 06, sprayed with underglazes, and final fired at Cone 5 or 6.

This project demonstrates the expressive potential of an emergent computational algorithm, different growth, adapted for clay 3D printing. I construct the machine toolpath by stacking each simulation step of the algorithm in Z-axis to construct the layers of 3D object. This method resembles the strategies of constructing 3D forms using 2D layers discussed in *Edge of Chaos* project in Chapter 3. Doing so embeds the temporal axis of growth-based algorithms in one of the spatial axes.

Although this form-making technique uses digital morphogenic principles, its clay fabrication falls back to a hylomorphic approach without 'joining forces' with the material [24]. A reason for this disconnect comes from the distinct workflows of the digital form-making and physical fabrication (for example, constant machine speed and extrusion). Constructing digital generative algorithms that utilize physical constraints as an emergent computational force would connect digital and physical form-generating methods more tightly. In the next section, I introduce another approach for constructing an emergent algorithm and clay fabrication processes that would provide better insights into virtual-material morphogenetic making.

5.3 Case Study: *Distributed Agency Trails*

In this project, I apply a similar layer-by-layer digital construction using a custom computational agent that mimics the location of the 3D printer nozzle in virtual space. I generate each point of the trajectory in an iterative loop as opposed to generating the whole layer. After a few layers of following a pre-defined geometry, the agent then attempts to follow its previous trajectory defined by parameterized attraction and sampling forces.

The design space of this project demonstrate variability in final results coming from the emergent behaviors of the computational systems (self-balancing, error accumulation, speed damping, etc.) and emergent qualities of the material (sagging, collapse, etc.). The post-production workflows like glazing and fixing the fabricated geometry demonstrates additional input into the design of the final artifact.

5.3.1 Digital System Design

The computational system builds on agent-based modeling paradigm implemented in Processing creative coding environment. Due to its pure symbolic environment, Processing offers less overhead compared to Rhino/Grasshopper environment, computationally performs better and allows for a smoother digital workflow.

The system contains one agent with `position` and `velocity` vectors that follows a circular trajectory in the initial layers (determined by an exposed parameter). After a set number of rotations, this agent starts following its previously constructed toolpath. As this tracking behavior unfolds over time, the agent's behavioral response to changes in various control parameters introduces path deviations, path reversal, and sometimes derailing from its path altogether.

The agent is constructed with the following parameters: the form's center coordinate

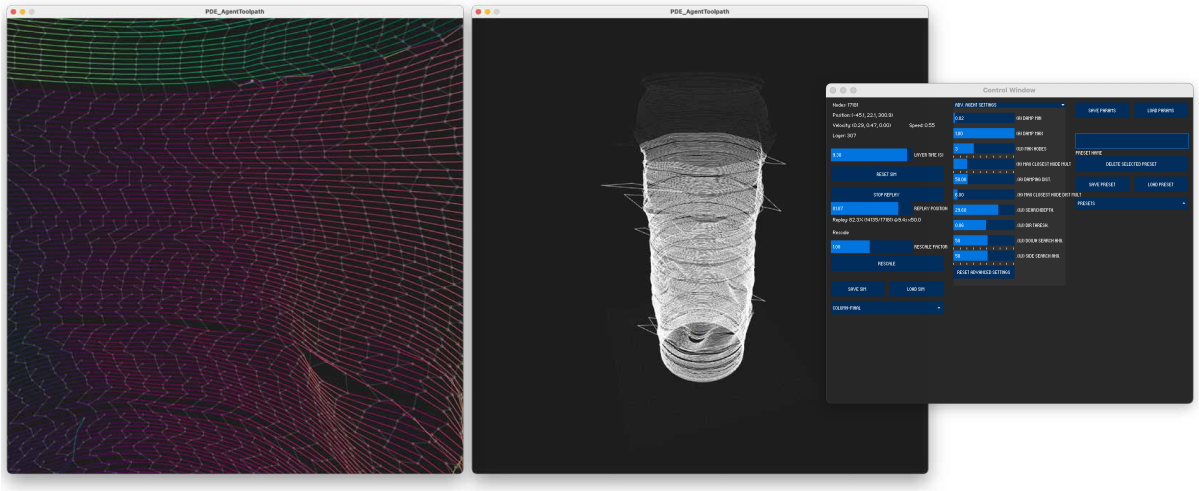


Figure 5.3: The interfaces of *Distributed Agency Trails*: (Left/Middle) The main visualization interface; (left) system is in generative mode and virtual camera is zoomed in, denoted by colored toolpath and network structure between layers; (middle) system is in the playback mode, denoted by the white toolpath. (Right) The parameter control interface.

`(sx,sy,sz)` and `radius`, the `initialLayerheight` for spacing the spiralized toolpath, and the `pointDistance` designating how far each vertex will be recorded.

Agent Trajectory Generating Logic

The agent follows its trajectory in a 3D space under the influence of three forces: The first is an intrinsic upward vector that spiralizes the agent toolpath. The second is a local damping to smooth rapid angular changes as the speed increases. And lastly, a pull force towards the previously traversed trajectory to steer the velocity vector towards the path. The update loop evaluates these vectors every frame, updates the `velocity` and `position` vectors, and inserts a new node whenever the traveled distance to the last node exceeds the `pointDistance`. This continuous trajectory can shrink or grow responding to the real-time parameter changes.

Shortest-distance Node Network

The agent's trajectory forms a temporally ordered list of nodes that corresponds to the spiralized toolpath. In addition, *Distributed Agency Trails* constructs a secondary graph structure on the fly to keep track of the vertical relationships between nodes in the toolpath (*network* label in Figure 5.4A). This graph contains the shortest distances under a certain node that is used for fabrication-related calculations.

When inserting a new node to the toolpath, the agent queries `N` nearest nodes that fall inside an adjustable conical zone and chooses one node with minimum Euclidean distance. This connection is stored as a link in `edgeOut` list of the `node` instance. This list contains both the network links and the temporal trajectory that forms the machine toolpath. These graph connections are recalculated upon exporting the GCode file or on-demand using its associated hotkey. Recalculation ensures the connections do not connect to wrong layers and produce correct extrusion amounts.

Parameterization

To control the system's evolution over time, many system parameters are exposed to a secondary user interface. A JSON-based `ParameterManager` class enables exploration of the design space by allowing save/load operations on the parameter states and by manipulating a group of parameters using presets. Primarily, `layerTime` parameter controls the agent's speed, producing more rugged toolpaths resembling surface textures with lower speeds, and smoother curves with higher speeds. One downside of using singular value for agent speed was the connection between spiralization and the layer height. Since the upward motion for spiralization (`vel.z`) is calculated by `initialLayerHeight / layerTime`, portions towards the top of the toolpath may have increased distance between two layers. This differential gets compensated during the fabrication process using

a variable extrusion calculations described later in Section 5.3.2.

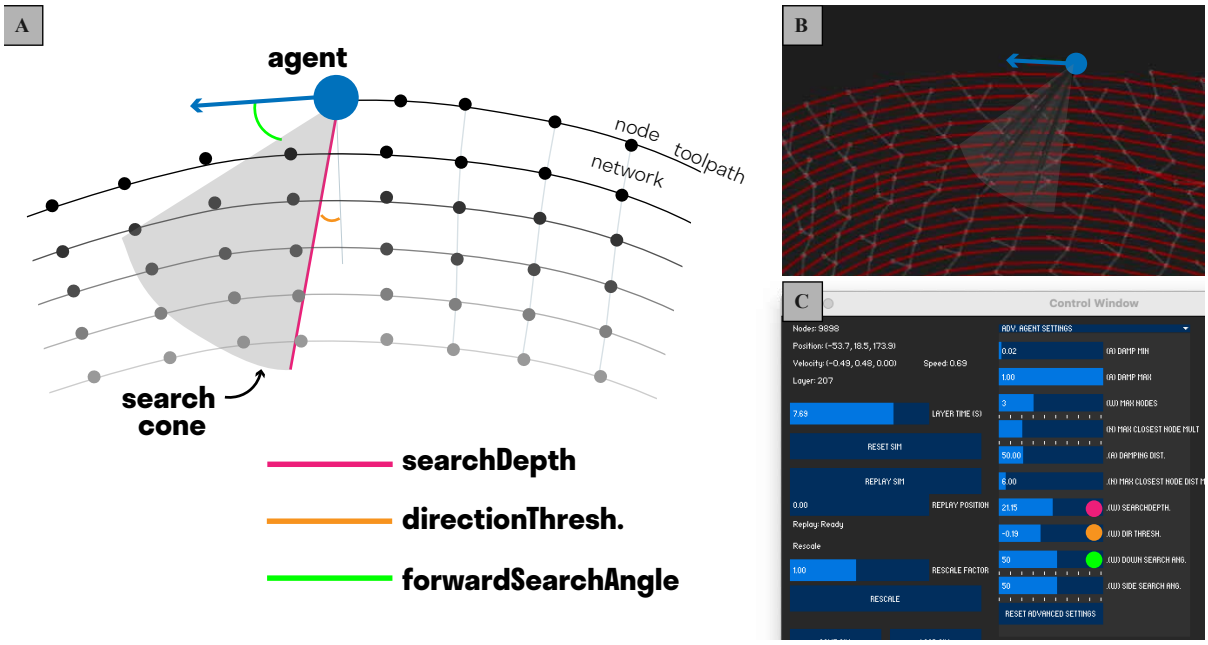


Figure 5.4: The parameter control mechanism of *Distributed Agency Trails*: (A) A diagram showing parameter mapping of the agent with labeled options; `sideSearchAngle` is omitted from the diagram. (B) Screenshot from the main Processing window with visualization; agent searches nodes within the search cone. (C) Second Processing window with all controls; colored circles marks the important parameters shown on the right.

The control interface exposes advanced parameters related to the internal mechanisms of path following. These can be categorized as parameters pertaining to movement damping, neighborhood search geometry, and weights for determining influence of other nodes' to the current behavior (Figure 5.4). Combinations of parameter values produce vastly different agent behaviors. For example, four parameters (`sideSearchAngle`, `forwardSearchAngle`, `directionThreshold`, and `searchDepth`) that control the search cone geometry effectively designate which previous nodes will be sampled for path following. When `searchDepth` is low, the agent often looses contact with the path and flies off; when this happens, increasing the value often returns the agent to the trajectory. The other three parameters control the shape of the search cone as shown in

Figure 5.4. The agent displays circular behavior when the `sideSearchAngle` is high. Similarly, when the search cone points in the direction of negative world Z (i.e. negative `directionThreshold`), the agent has a higher likelihood of flipping its direction (direction flip is visible on the toolpath visualization in Figure 5.5A).

Other UI Features

The trajectory is visualized using the calculated normal vectors for each node. This is especially useful when the agent changes direction and starts traveling the previous path backward. The agent, its movement properties such as direction and sampled nodes, the nodes themselves, and network connections can be visualized to support better understanding of the internal system behavior.

In addition, the user interface contains a JSON-based simulation state recorder. Since every run of this computational method produces a completely different form, this feature enables building on or modifying a previously generated form. This module saves the simulation state with `metadata`, `agentParams`, and an array of `nodes` in a local JSON file to be loaded at a later time. The system allows resuming the simulation at the save time or deleting a desired number of layers to return to a previous layer.

The user interface also allows for replaying the toolpath by interpolating between stored nodes with a parameterized index and speed. A slider allows for scrubbing the currently played position, and the `layerTime` parameter changes the speed of this playback when used in replay mode. This preview is visualized with a reduced opacity for the full path and with solid strokes for the already-previewed portions.

Borrowed from *ClayToolkit*'s reexport feature, I also implemented a rescaling feature for the toolpath. This allowed for uniform scaling the finished form without needing to run the emergent simulation again.

5.3.2 Physical Production Workflow

Distributed Agency Trails uses two physical mediums, clay 3D printing and 2D pen plotting, to fabricate digitally generated toolpaths. For clay 3D printing, I adapted the fabrication components described in Section 4.2 to create the GCode file. I did not spiralize the toolpath because the digital simulation already ran in a spiral format. A major upgrade to the GCode generation was continuously modulating material extrusion rate and machine speed. And, for pen plotting, I exported 2D SVGs from the 3D toolpath in the digital system and used a commercial software to convert the vector file into plotter executions.

Variable Material Extrusion and Machine Speed

The adapted GCode export module in *Distributed Agency Trails* takes in the *network* of nodes containing agent coordinates and their connections to other nodes. Using this information, the system calculates the horizontal distance between two points for extruding the material (same as the other systems discussed thus far). In addition to this, it also calculates the vertical distance between the layer underneath the current point. Instead of using a fixed layer height, I use this vertical distance to calculate a variable amount of material at each point. For example, if the agent had run off and left a gap in the toolpath, the layer that travels on top of this gap extrudes more material since the vertical distance is greater than the original layer height. Similar to the variable extrusion, I use this metric to slow down the printer in less supported areas. This is due to the clay's behavior during 3D printing where changing from tightly compressed regions to loosely supported parts tend to break the extruded line and introduce weak points in the final form.

These two modifications to the GCode generator enabled stronger bonds between

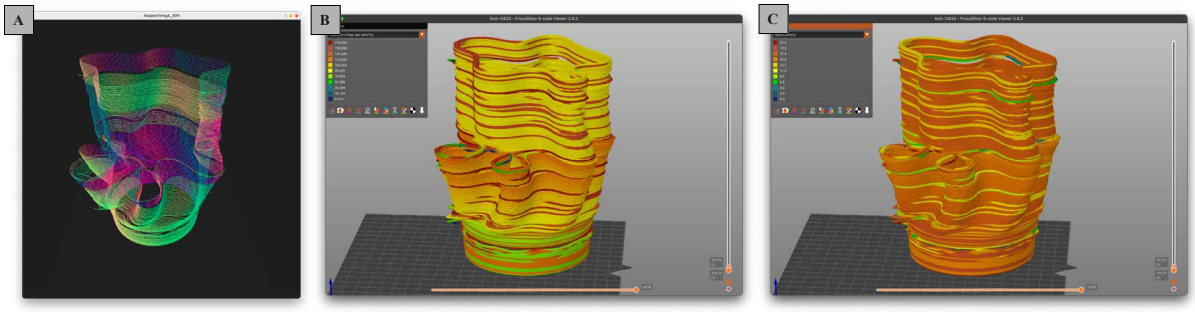


Figure 5.5: Toolpath visualizations. (A) The toolpath within the generative system; toolpath is visualized with colors based on the path normals. (B-C) Visualizing the GCode in Prusa software, showing volumetric material deposition and printer speed, respectively.

layers and produced significantly more stable objects once the print was complete.

Ceramic Fabrication

I fabricated six clay 3D printed objects enabled by *Distributed Agency Trails* in May 2025. I used a recycled Nara 5 clay body on a PotterBot Micro with 4 mm nozzle for fabricating the physical objects. Two days prior to printing, I put a rag inside the plastic bag of clay and added 250 mL of water. I then wedged this soft clay, hand loaded and assembled the tubes, and loaded on the machine.

During clay 3D printing, I noticed how much difference the variable extrusion makes in print quality. The usual workflows of manually compressing some layers was not needed. Most prints succeeded in the first try, with the exception of the object shown in Figure 5.6. Using a sponge as a manual support in the second try, I fabricated this object two times after first collapsed print (Figure 5.8A). In order to make the material qualities explicit, I printed the column-like object shown in Figure 5.12 twice with the exact same toolpath. Even though the printer followed the exact path and extruded the same amount of material, the surface details look different in certain areas. In addition to emergent material qualities, I also noticed an emergent soundscape produced by the



Figure 5.6: Toolpath creation and clay fabrication workflow one ceramic artifact. (A) The toolpath is being constructed by the agent visible with black lines at the top; toolpath is visualized with colors based on the normal vectors (i.e. alternating colored regions imply agent flipped movement direction). (B) Object is being printed on PotterBot bed. (C) Applied glaze on bisqued object; inside the kiln before final firing. (D) Finished ceramic object.

printer motors due to continuously varying the extrusion and speed. The modulation of the printer execution was reflected in an emergent sonic composition in the sounds of the 3D printer's motors for XY-axis, Z-axis, and extrusion.

Once the objects were dry after a day, I manually attached two of the small objects together when they were leather-hard. I then bisque fired all at Cone 05 with 3 h preheat. I used longer preheat times (keeping the kiln temperature between 82 °C to 93 °C) to remove the remaining moisture and prevent cracking during the full firing cycle. I refrained from using glazes for most objects to show the interactions between the digital toolpath and the printed material. However, I applied small spots of a black crater glaze to two objects. I then fired the pieces to Cone 6 with 1 h preheat and 20 min hold times. Although the glaze spots were flat before firing, the crater glaze produced beautiful swelled areas with a rough texture after final firing (Figures 5.6C and D).

After final firing, one object was stuck on the kiln shelf that was not covered in kiln wash. As I removed this object from the shelf, it chipped around its base. I needed this

object to be visible in an exhibition, so I used a workflow that contains a precise FDM printing with PLA to create a base.

Fixing Broken Base with FDM Printing with PLA

I approached this breakage as an opportunity to demonstrate the affordances of other materials and fabrication techniques that juxtapose the unpredictable nature of clay. I used a Bambu Labs X1 Carbon FDM printer with PLA material to fabricate an attachment to fit in the cracked region and create a flat platform for the object to stand upright. In order to fabricate this object precisely, I used Polycam 3D scanner app that utilizes the LIDAR of iPhone. Processing 20 pictures around the broken base gave me a good quality digital model. I then imported this in Rhino, extracted the triangle faces aligned with world Z-axis in Grasshopper, cleaned the resulting selection manually to remove stray faces, and extruded this region to make it taller. Once I had the model, I iterated twice to match the digital dimensions with the physical measurements by rescaling the object. And the second version fit inside the cracks perfectly and allowed the object to stand upright without gluing the PLA piece to the base. This workflow is documented in Figure 5.9.

Pen Plotter Fabrication

Distributed Agency Trails also used another material medium for demonstrating the digital aspect of the toolpath generation. I used an Axidraw V3 pen plotter to plot generative toolpaths from various perspectives. I used various-sized black fine-liner pens on a white 67lb/147gsm paper.

In the digital system, I set up the virtual camera in two positions for each object, one directly from the top, and the other is angled on the side (Figure 5.7). Keeping the perspective projection allowed me to capture various details of the toolpath even

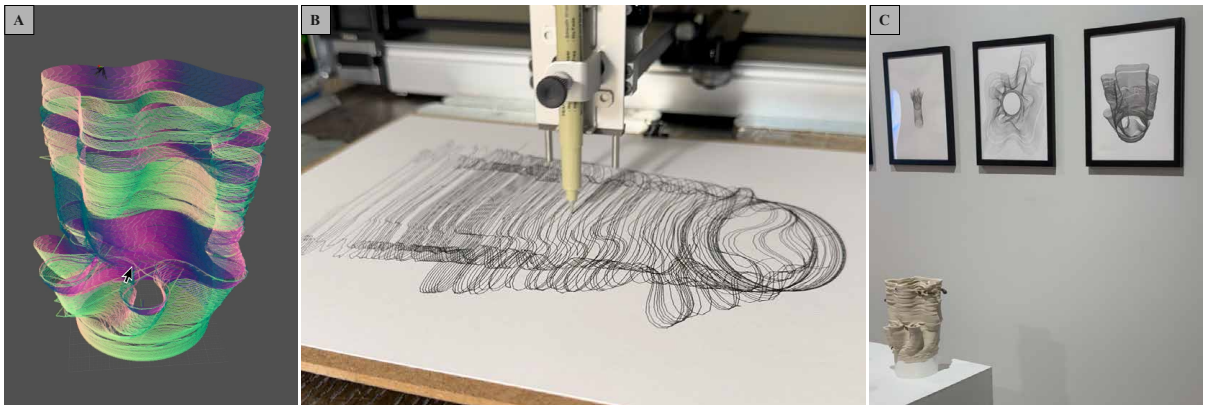


Figure 5.7: Pen-plotting workflow based on the generated toolpath. (A) The generative toolpath was saved as SVG. (B) Pen-plotter leaving 0.25 mm marks as it follows the toolpath. (C) Framed plotter piece in the installation along with ceramic artifact.

from the top view. I then exported the toolpaths as SVGs from these perspectives, with variations for enabling or disabling the visualizations of the network points. I then used the Inkscape plugin for the machine to convert these SVGs into plotter instructions.

Through developing an emergent computational process, ceramic fabrication, and pen plotting, *Distributed Agency Trails* demonstrates the use of various creative agencies in the same exhibition. It demonstrates control mechanisms for the digital interface, extrusion modulation based on the digital geometry, and representing the digital toolpath with physical plots with different pens. This method of constructing hybrid artifacts resonates both with digital morphogenesis principles *and* physical morphogenetic making.

In the next section, I introduce another prototype system that uses various creative agencies in a large-scale media installation. This project makes use of the digital simulations of the *Edge of Chaos* described in Section 3.2 and physical artifacts of *Morphogenic Clay Vessels* described in Section 5.2 in conjunction with other computational, physical and biological processes to create a cyber-physical ecology of distributed agencies.

5.4 Results

This section presents the outcomes that emerged from the systems described in this chapter. These results provide evidence for the specific ways computational and material agency manifested in the final objects. To illustrate this, I present detailed visual documentation and experiential anecdotes that delve into the workflows involved in producing certain artifacts.

5.4.1 Material Glitches and Breakages

I have encountered various unexpected behaviors in material production, arising from interactions between different materials and environments. These unexpected behaviors manifested as glitches or material failures, such as breakages. A subset of them has been discussed in Section 4.5, focusing on digital solutions to manual problems and manual solutions to digital problems. Material glitches and breakages are central to material engagement and serve as core evidences for material agency.

Some of these behaviors are linked to clay's plastic state when wet and brittle state when dry. Collapse is a reality of clay 3D printing, and ceramicists approach this by making fast clay supports, or using sponges to support the weight of the clay. For example, as I printed the object shown in Figure 5.6, my first attempt resulted in a collapse primarily due to poor extrusion at the initial layers (Figure 5.8A). I recycled the clay and started another print, paying close attention to the initially failed region and using a sponge to support the object. Similarly, clay objects can become extra brittle when bone-dry. Figure 5.8B shows a large result of *Morphogenic Clay Vessels* printed with under-extrusion, causing insufficient layer compression. My attempts at trying to fix issues (left) only resulted in crumbling into pieces (right). This experience prompted material volume experiments explained in Section 4.2.2 and enabled a digital integration

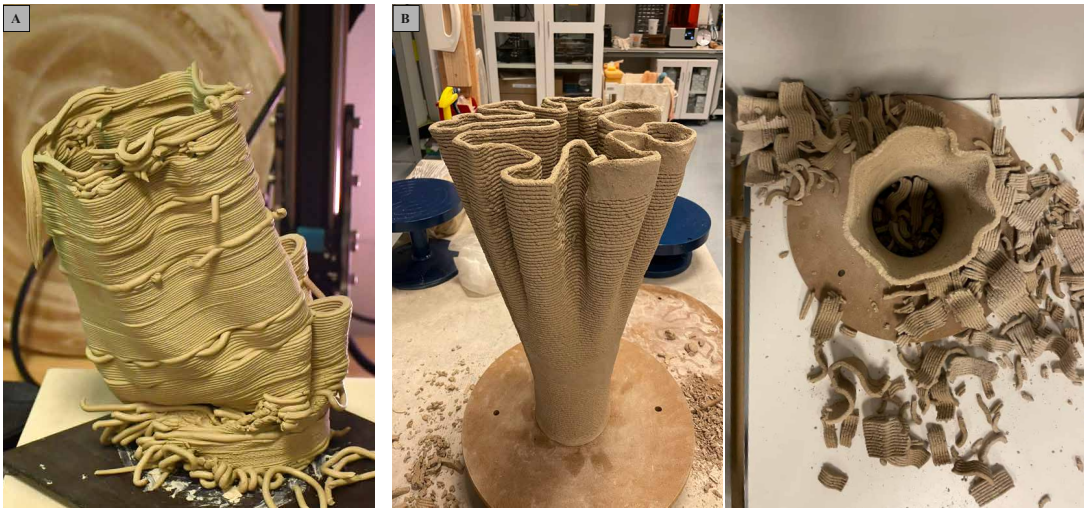


Figure 5.8: Examples of collapses and breakages. (A) The first attempt at clay printing resulted in a collapse. (B) Poor layer adhesion creating brittle objects when dry; attempts at salvaging the large print resulted in a breakage.

to address extrusion issues. I printed a similar size object again using two tubes of clay (Figure 5.13B). This humbling example is a stark reminder to what might happen, as Ingold says, if not 'joined forces' with the material [24].

Other behaviors originate from material chemistry and their interaction with extreme temperatures of kiln firing. For instance, although Aardvark[1] advertises the SBFG clay body as a buff-colored clay, we observed green-colored artifacts after final firing. As we did not continue using this clay body, I do not have additional data points to reason about this behavior. Similarly, immersed in high temperature setting inside the kiln, complex glaze chemistry might behave unpredictably, and some ceramicists, like our collaborator Raina Lee, specialize in glaze chemistry. When I used her black crater-glaze recipe thicker than I should have, vitrified glaze swell up to roughly four times to its expected size (lower-left object in Figure 6.2). I later used this as a material feature and created puffy dots around one of the outcomes of *Distributed Agency Trails*, shown in Figures 5.6C and D. These anecdotes resonate with what ceramicists call 'kiln gods' to

[1] A well-known clay retail store in California.

refer to the unpredictable nature of kiln firing.

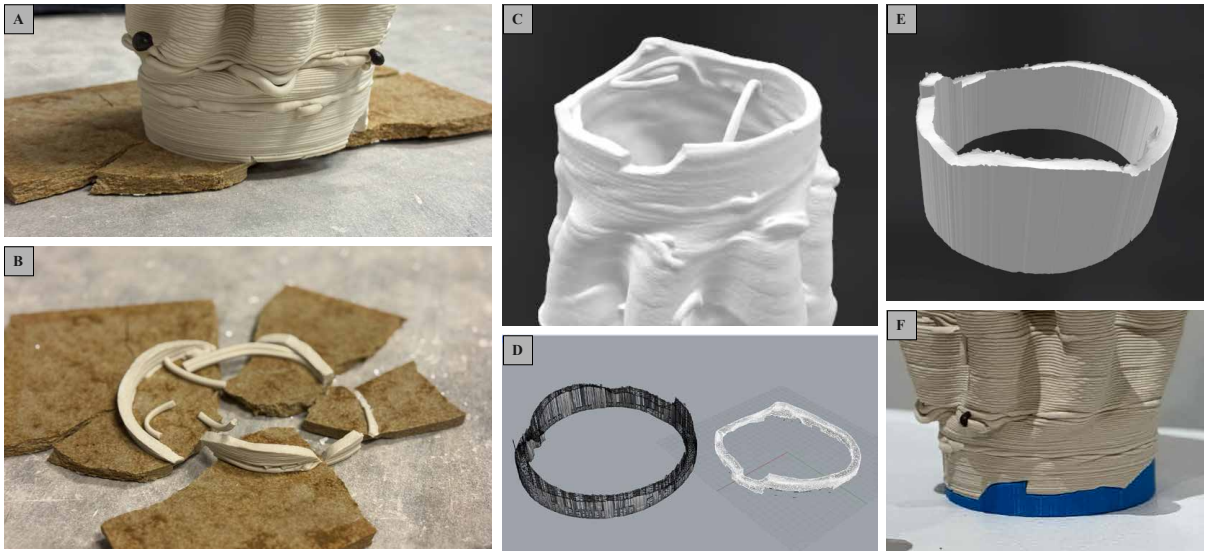


Figure 5.9: The repair workflow for chipped artifact involving highly precise LIDAR scanning and FDM printing. (A) Artifact was stuck on the bare kiln tile and (B) chipped as I removed separated them. (C) 3D model after digital LIDAR scan. (D) Extracting bottom faces in Rhino and fixing the mesh. (E) 3D model of the reconstructed base. (F) PLA-printed base fits after two iterations.

During the production of these results, I also observed other materials that juxtaposed clay's behaviors. While I was producing pen plotter pieces in *Distributed Agency Trails*, I found that the dense toolpaths produced solid black results with thicknesses larger than 0.3 mm. In one instance, plotting the small object's toolpath (third frame from the right in Figure 5.14A) created a challenge, since even the thinnest pen with 0.2 mm tip produced a dark blob. As I noticed this pen's ink was running out, I re-plotted the same toolpath and achieved a legible piece. This showed how ink's behavior can be a nonhuman collaborator in hybrid workflows. Moreover, Figure 5.9 shows a visual documentation of the steps of LIDAR scanning the broken portion and FDM printing with PLA. As described in Section 5.3.2, high-temperature kiln environment also made an artifact stuck to an unwashed kiln tile, and the base of the object cracked as I tried removing from the tile. I approached this breakage with another material workflow that

shows high-machine precision and juxtaposes the imprecise nature of clay.

Most of the results discussed in this subsection would look dramatically different at the hands of more experienced craftspeople. However, I find creative conclusions among these unexpected behaviors similar to the emergence of 'happy accidents' in generative art domain. Digital artists use this term to signify something going wrong in their code but the unexpected result is a favorable outcome rather than a disliked one.

5.4.2 Emergent Outcomes Afforded by Generative Systems

Here I present some results made possible by the generative algorithms *Morphogenic Clay Vessels* and *Distributed Agency Trails*. The behaviors of these computational systems and their parameter modulations produced geometries different from the ones showed in Chapter 4.

The implementation of multiple object growth feature in *Morphogenic Clay Vessels* produced groups of objects that respond to one another. Figure 5.10 shows examples of this method that fits into one another like puzzle pieces. I used the mapping function shown in the system description for the `variableHeight` parameter. After clay printing



Figure 5.10: Three vessels digitally grown together by avoiding each other. I printed each object separately and attached bases with uneven clay to keep them upright.

these objects separately, I attached a base by adjusting the amount of clay inside the vessels so that their center of gravities would sit on top of their bases. This modifica-

tion requirement foregrounded the distinct realities of digital generativity and physical production.

Figures 5.11A and C show results of spatially modulating the growth parameters. In (A), I assigned a solid geometry to the sides of the toolpath in Rhino viewport to block the growth for the vertices falling inside this geometry. In (C), I combined growing and shrinking regions by modulating the `separationCohesionRatio` parameter in the same toolpath. These examples display varying growth patterns in their construction. Similarly, Figure 5.11B shows application of displacement-based surface textures on a

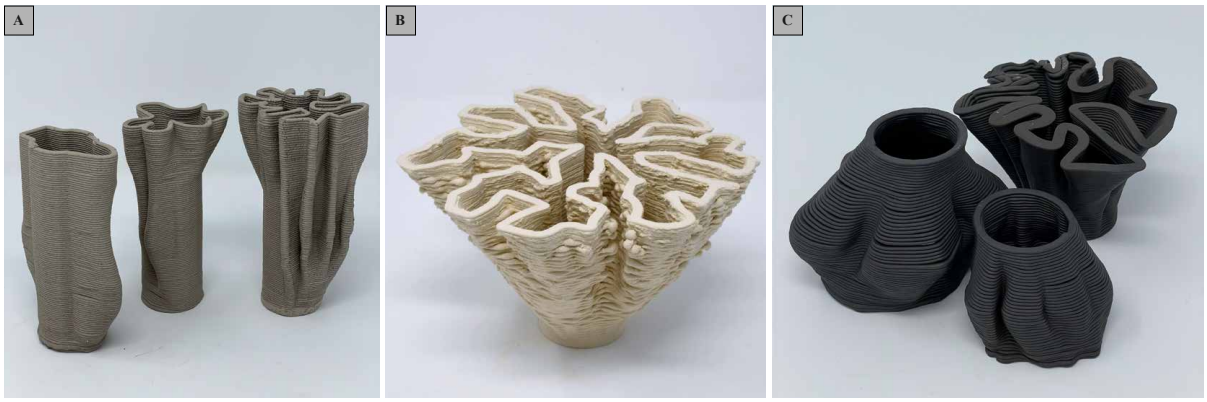


Figure 5.11: Initial results from *Morphogenic Clay Vessels* system demonstrating different control strategies. (A) A group of objects fabricated using growth blockers producing localized control around the vessel. (B) Fully-grown vessel with displacement-based surface textures that shows the 'age' of vertices as texture strength. (C) Small objects containing growing and shrinking regions in the same toolpath.

generative form. In this case, the texture strength shows the age of the vertex (i.e. vertices in the original starting shape gets a larger bump at the top).

In *Distributed Agency Trails*, initial smaller objects did not contain enough spatial and temporal resolution to show the agent's emergent behavior. Designating a larger starting radius and allowing the system to unfold for longer unlocked richer form and surface articulations. This resonates with the notion I discussed in Chapter 3 that increased number of interactions between system component resulting in more complex

behavior. I increased the starting radius and started experimenting with the advanced parameters described in Section 5.3.1. After many iterations in system behavior and parameter configurations, I started tuning in the effects of advanced parameters and started observing the most promising results. Once I tuned its parameters for larger dimensions, I used the system to generate two large toolpaths (Figures 5.6 and 5.12). Printing the

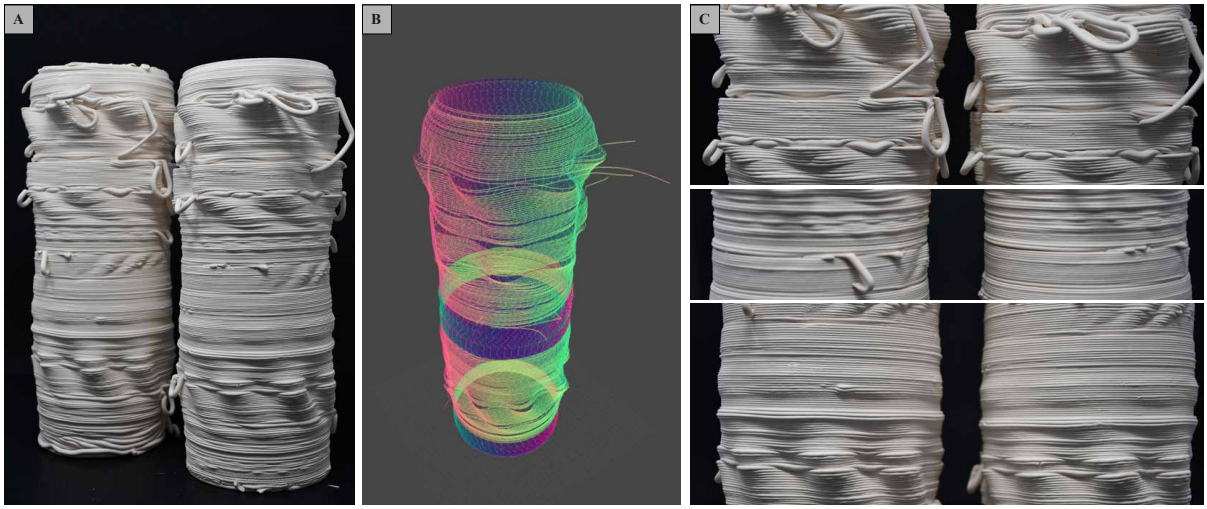


Figure 5.12: Side-by-side comparison of the surfaces of objects fabricated by an identical toolpath: (A) Two column-like vessels. (B) Toolpath used to fabricate the objects. (C) Close-ups of various locations on their surfaces showing micro variations.

identical toolpath shown in Figure 5.12B twice produced visibly different micro-features on the surface of the vessel. These differences on the physical vessels evidence material contingency despite a fixed GCode. This divergence adds local surface variations without compromising the global form and demonstrates a version of material agency in action.

5.4.3 Presentation Context

I exhibited these pieces in group exhibitions in 2022 and 2025. The outcomes of *Morphogenic Clay Vessels* was shown in *SyMADES* exhibition (Figure 5.13) as a part of demonstrative examples of the artist residencies. Conversations with audience members revealed that the computational and life-like qualities of these ceramic objects was



Figure 5.13: Various outcomes of *Morphogenic Clay Vessels* were exhibited at *SyMADES* in 2022. (A) Small objects were labeled by their clay bodies (left-to-right, Bone White and Obsidian) on pedestals at UCSB Elings Hall. (B) Large differential growth vessel (printed with FSB clay body) was displayed along with smaller vessels at SBCAST.

visible from its construction. An unexpected reaction was the general interest in the interior surfaces of these objects. Based on this reaction, the future iteration of using these geometries will create holes or slice the object vertically that reveal their internal structures.

I exhibited the artifacts produced by *Distributed Agency Trails* at *DeepCuts* exhibition along with an explainer video that describes the generative approach and pen-plotted pieces (Figure 5.14). The pen-plotted pieces demonstrated the adaptable nature of the generated toolpath to multiple media. Audience members expressed that the pen-plotters made the generative nature of this work more clear.

The outcomes produced by this chapter's projects resemble natural patterns or produced by computational agencies. Based on this artistic context, I used diverse outcomes produced by *Morphogenic Clay Vessels* system, and a few more ceramic vessels in *Simulacra Naturae* exhibition at UCSB Elings Hall (Figure 6.2). This chapter omits discussing *Simulacra Naturae* in the context of distributed creative agencies to avoid heterogenizing its focus on generative computational systems and material agency of fabrication. Its



Figure 5.14: Various outcomes of *Distributed Agency Trails* were exhibited at *Deep Cuts* in 2025. (A) Five ceramic objects, their pen-plotted toolpaths, and a explainer video were displayed at UCSB Elings Hall. (B) Same ceramic artifacts and a selection of the plotted pieces were shown at SBCAST.

connection to this dissertation's central concern is elaborated in Section 6.2.1 along with future development plans.

5.5 Discussion

This discussion reflects on the opportunities, challenges, and limitations of using generative computational methods within digital clay fabrication. The primary insights come from the two case studies, *Morphogenic Clay Vessels* and *Distributed Agency Trails*, and are situated within the broader trajectory developed across Chapters 3 and 4. These

projects enact emergent expression by making artifacts arising through distributed creative agency across human, computational, and material actors.

5.5.1 Real-time Digital Interaction for Physical Production

Across *Morphogenic Clay Vessels* and *Distributed Agency Trails*, real-time digital interaction emerged as a prerequisite for discovery in generative making, even when physical fabrication proceeds asynchronously. As discussed in Chapter 3, liveness in algorithmic systems enables rapid parameter exploration and the perception of emergent tendencies that only become legible through fast and instrumental iteration.

In *Morphogenic Clay Vessels*, scaling the artifacts demanded an increased number of vertices to achieve the required curvature for undulations. Implementing the system in GhPython within Grasshopper introduced significant performance overhead during toolpath generation: each parameter change incurred 15 s to 60 s of processing. Batching parameter edits mitigated latency but constrained the fine-grained exploration needed to dwell near the edge of criticality in the configuration space (Chapter 3). By contrast, *Distributed Agency Trails* was implemented in Processing, a creative coding environment better suited for real-time rendering. Despite being CPU-bound, the system comfortably generated and displayed more than 20,000 vertices at interactive rates (60 fps). This fluid responsiveness changed the quality of engagement and enabled me to explore parameters freely, discover emergent forms, and collaborate with the custom algorithm.

Although neither system fully leveraged GPU computation, both make clear an asymmetry central to hybrid practice: when digital emergent forms are coupled to physical production, the cost of a failed physical iteration is high, whereas digital iterations are comparatively cheap. As detailed in Chapter 4, clay processes are labor-intensive, risk-laden, and time-bound, therefore, investing in digital liveness enables better preparation

for physical fabrication. Over time my clay expertise (informed by work with professional practitioners) improved, refining the mapping between extrusion behavior and toolpath dynamics (Chapter 4). This maturation enabled a deliberate trade-off: reduce computational over-specification where the process-driven complexity of clay can productively take over, while retaining digital precision where structural safety and machine coordination demand it.

5.5.2 Tensions between Digital Precision Expectations and Material Realities

My background inclines me toward constructing precise computational solutions. Yet active materials often resist such approaches: much of the digital clay workflow necessarily engages material processes that are imprecise by nature (Chapters 4 and 5). The practice therefore centers on negotiating control and autonomy across digital and material agencies, rather than eliminating variability.

Formative experiments in *Morphogenic Clay Vessels* experienced this tension. A morphogenetic and CAM-driven toolpath generation method hinted at a non-hylomorphic approach to form-giving [24], yet in practice it sometimes enforced an external order upon the clay. This disconnect prompted toolmaking that solved immediate production issues (e.g. PotterBot extrusion calibration; see Section 4.2.2) while also forcing a reassessment of my pursuit of 'perfection' within an inherently 'vibrant' craft [11]. In *Distributed Agency Trails*, the desire to reveal material properties led me to control extrusion per vertex. This decision surfaced latent potentials as stable scaffolds and distinctive surface articulation, but also re-centered digital control, and by extension the human agency authoring it. The productive tension lies not in choosing one pole over the other, but in continuously re-calibrating the roles in this collaboration. In other words, deciding

where to let matter speak and where to constrain it so the form can survive fabrication.

A similar negotiation appeared in the plotting studies. SVG-driven machine trajectory reflected digital representation perfectly, and the ink appeared compliant, until it began to run out. In this context, the 'imperfection' became a resource while plotting the toolpath of small objects, in which ink depletion produced lighter strokes. This material glitch felt aligned with material contingencies as co-authors of form.

Taken together, these tensions locate the site of expressivity in my hybrid practice in neither purely computational nor purely material domains, but in their ongoing negoations.

5.5.3 Limitations

The systems presented here have not been evaluated extensively by professional craftspeople. Collaborators in residencies brought established formal languages that do not necessarily require emergent digital forms. For instance, Pilar's multi-headed saddle forms or Eun-ha's narrative figurative work (Figure 4.1). Moreover, the *Morphogenic Clay Vessels* prototypes were not sufficiently responsive for on-the-spot, iterative exploration during residencies. Future studies could prioritize co-design protocols with expert ceramicists, focusing on where emergent digital methods augment rather than replace existing practices.

Algorithmic and material constraints remain. Differential growth effectively fills space without self-collision, but physically manifests as rapidly increasing material usage at upper layers, yielding heavy artifacts and requiring tube swaps in advance. Next iterations could incorporate material-aware objectives (e.g. penalizing local infill density or modulating bead width) to shape emergent forms toward lighter structures. In *Distributed Agency Trails*, the starting layer is restricted to a circle. Initial experiments with ar-

bitrary SVGs produced inconsistent agent dynamics, as sharp corners induced abrupt velocity changes. While damping partially addressed this, a more robust approach is to impose curvature and acceleration constraints on initial conditions (e.g. spline-based smoothing with arc-length parameterization) so that growth begins from dynamically coherent states. From a systems standpoint, CPU-bound implementations limit scale and fidelity. Migrating core routines to the GPU would support larger agent populations and richer local interactions at interactive rates, further amplifying the liveness discussed above. Future work could also explore multi-agent ecologies that interact, compete, or average to compose a single toolpath, as well as live, interactive printing wherein sensing feeds back into growth during fabrication.

Collectively, these directions aim to better align computational emergence with the realities of clay, and to stage collaboration between human, machine, and material agencies at the tempo of making.

These case studies demonstrated how creative agency within hybrid making can be redistributed among human, computational, and material actors. These projects indicate that hybrid emergent expression relies on the ongoing negotiation among these human and nonhuman actors.

The upcoming final chapter reiterates the results of the three chapters cohesively, discusses their implications in the broader space of new media research, and concludes with avenues for future research.

Chapter 6

Conclusion

To close this dissertation, I synthesize how six projects across three chapters position distributed creative agency in my practice. I then interpret these findings theoretically, clarify the roles of human and nonhuman agencies, revisit the research questions, and outline limitations. The chapter concludes with the full prose for *Simulacra Naturae* and directions for future work.

This dissertation presents findings from my art practice and practice-led HCI research conceptualized as *emergent expression*, combining nature-inspired aesthetics in computational complex systems and craft-based digital fabrication research. In this hybrid practice, I distribute my creative agency with digital generative systems and active craft materials like clay, and negotiate artistic intent and control. To demonstrate the computational agency present in generative algorithms, results in Chapters 3 and 5 showed various outcomes of artificial life simulations and a custom algorithm. To demonstrate the material agency present in clay fabrication, results in Chapter 4 showed unpredictable behaviors seen with regard to clay shrinkage and deformations. Chapter 5's results section demonstrated such behaviors with regard to clay plasticity and drying workflows and high-temperature kiln interactions, and juxtaposed other materials' behavior, like

ink and PLA, with clay. To demonstrate human agency and positioning my creative intent, I present how I approach producing these results by engaging with parameter space exploration and developing fabrication techniques that reveal or constrain material behavior. These approaches took place within the development of novel computational tools, techniques, and workflows.

6.1 Discussion

This section reflects on what the results mean for *emergent expression*. I articulate a theoretical reading of the work, discuss how agency circulates across human, computational, and material actors, and surface implications for practice. I then revisit the research questions and outline limitations to situate the contributions.

6.1.1 Theoretical Interpretation of Emergent Expression

A significant motivation in this research is understanding various art practices and modalities as abstracted systems containing interacting components, similar to complex systems. This motivation comes from an interest in understanding the non-linear world around us. The source of this non-linearity lies within various complex systems in nature, and studying them can also reveal insights into human expressive complexity. As media artist and researcher Memo Akten expresses through his self-organizing artworks, studying the nature of nature through complex systems and morphogenesis can shed light on human expression and how this relationship is mediated through technology [226].

Primary tensions between digital and physical domains create an opportunity for investigating making activities with a more abstracted framework. When investigated with a complexity perspective, this hybrid creative process contains relational networks between various actors in the forms of software, materials, tools, hardware, and artists, and

the expressive outcome emerges from the distributed agency in the form of information flow.

This dissertation adopts a complexity lens to understand art-making as an ecology of interacting actors across digital and material domains. This lens extends Galanter's complexity-inspired generative art theory into hybrid domains and acts as a generative notion rather than a quantitative complexity science research [25]. Following this view, creative outcomes are not the product of any single agent but emerge through relations among software, materials, hardware, environments, and people. Hoelscher characterizes the art world as an information ecology in which "art is information in its most active, open, peculiar, and irresolvable mode" [68, p. 16]. Irresolvability here names the continued unfolding of complex systems and sustained generativity that emergent practices cultivate.

I treat creative agency as distributed and relational rather than intrinsic to any single node. This framing invites a shift from describing *what* individual actors are to analyzing *how* agency circulates among them; in other words, how information and influence flow across the network. Thinking about practices as relational networks helps identify the interactions that give rise to emergent form and behavior. This also illuminate practices across virtual and physical domains, where digital simulations and material processes co-produce outcomes through feedback loops, constraints, and affordances that are legible in systems terms.

Under this theoretical framing, *emergent expression* affords an abstraction through reasoning about practices at the level of relations and flows, a translatability through transforming insights across media by focusing on interaction patterns, and a sustained novelty through designing for open-endedness with unfolding processes rather than closure with fixed-forms.

6.1.2 Agency, Unpredictability, and Control in Hybrid Practice

Human Agency

Throughout this dissertation, I have presented scenarios in which human, computational, and material agencies intersected to shape formal properties of an artifact. This naturally raises the question: whose creative agency is being supported? This dilemma, whether my systems primarily extend my own artistic practice or support the creative intentions of others, remains central to my reflection and warrants further inquiry.

The human actor within this distributed network of agencies is, first and foremost, myself. My artistic intent has been to foreground life's interconnectedness, relational structures, and the situated role of the human within these networks. This orientation resonates with perspectives in complexity-informed art practice, where artists work not to impose form but to reveal emergent dynamics across multiple interacting agents [25]. In Chapters 3 and 5, I illustrated this position through generative algorithms and/or fabricated artifacts that enact such relational structures and their emergent aesthetics.

Yet, the case studies in Chapter 4 demonstrate another orientation: I developed systems explicitly to support the workflows of skilled craftspeople. In this context, the focus shifts from my own expressive goals to those of collaborators, aligning with the HCI tradition of creativity support tools [173]. However, these tools not only amplified the agency of others by facilitating design and fabrication of artifacts, but also expanded my own practice into material domains. By engaging in sustained collaboration, I acquired tacit knowledge of clay's behaviors.

This became particularly evident in my later project, *Distributed Agency Trails*, where my material sensibilities were more finely tuned than in earlier efforts such as *Morphogenic Clay Vessels* and allowed me to 'join forces' with the material [24]. Supporting others thus functioned as a pedagogical direction for myself: through building systems that

scaffold other's creative agency, I also deepened my own tacit understanding of material and process. Seen this way, my work negotiates between supporting my agency through computational systems and supporting the agency of others through tool design.

Conceptualizing this dynamic through ANT reframes the dilemma itself: Agency is not a property that belongs to me, my collaborators, or the software, but an emergent effect of the relations among all involved actors [201]. From this perspective, the question is less about whose agency is supported, and more about how agency is redistributed and reconfigured as human, computational, and material actors 'intra-act' [188]. My research thus underscores emergent expression as a distributed phenomenon: one that does not privilege a single author but emerges through the shifting alignments of heterogeneous actors.

Unpredictability, Learning, and Discovery

Emergent expression offers a space to collaborate with outcomes that are difficult to predict and often unexpected. This is one of the central reasons I am drawn simultaneously to dynamic complex systems and to clay fabrication: both domains afford cycles of exploration, learning, and discovery. While the dynamics differ across physical and digital contexts, both foster forms of knowledge that arise through sustained engagement with unfolding processes.

Extended engagement with clay as a material generated a growing familiarity with its behaviors. This aligns with the way expert craftspeople cultivate tacit knowledge over decades of practice, gradually attuning to how materials respond to environmental variables, processes, and interventions. Through *learning* from collaborators and sustained material practice, I began to experience clay's processes as less surprising over time. This was particularly evident in the fabrication of forms generated with *Distributed Agency Trails*, where production proceeded smoothly and I could respond to acute breakdowns

and correspond with material. Generative algorithms, by contrast, operate as instruments for *discovery*. Artists learn the emergent behaviors through iterative parameter modification and careful observation of the system's unfolding dynamics. In projects like *Edge of Chaos* and *Distributed Agency Trails*, real-time parameter exploration allowed me to treat algorithms as responsive instruments: sustained practice revealed insights into which variables were worth manipulating, which produced predictable effects, and which remained stable across contexts.

The most striking difference between digital and material domains seems to be the *permanence* of what is learned. In computational systems, discoveries about emergent behaviors remain valid as long as the underlying system logic does not change. The mapping between parameters and behaviors persists, though it may shift radically when new rules or adaptive learning mechanisms are introduced. For example, experimentation with Physarum simulations in *Edge of Chaos* revealed the agent movement parameters as the primary driver of emergent forms. If I modify the agent behavior logic to a learning-based method, I would need to learn new set of evolutionary parameters to control their emergent behaviors. In material systems, however, it seems as though the discoveries accumulate: as a craftsperson learns how a certain geometry of one clay body shrinks or interacts with specific glazes, this knowledge may transfer to future iterations of the practice. For example, as production potters make repetitions of the same vessels and glaze combination, they accumulate knowledge regarding the material behavior throughout the ceramic pipeline.

Even though I believe this is the case, I have observed discussions about irreducible material unpredictability in the interviews with expert craftspeople (for example, frequent references to 'kiln gods' to refer to the unpredictable nature of firing). This might be tied to *complexity* and *risk* factors. The number of component interactions driving the emergent behavior differs between two domains. While computational systems use an

abstracted version of real-world complex systems sufficient to display emergent phenomena, physical production operates in its full situated complexity. Increased component interactions may display unpredictable behaviors with minuscule differences to workflow steps. In addition, risk factors differ between the domains. In physical domains, mistakes in any part of the pipeline may erase weeks or months of labor, whereas mistakes in computational implementations do not carry effects into the consecutive executions. These points may explain why craftspeople mention unpredictability even after decades of practice.

6.1.3 Revisiting Research Questions

In addition to addressing the research questions in the beginning of each chapter, I briefly revisit how the projects described in this dissertation answer the research questions:

RQ1. What processes and procedures in computational and material domains enable designing with morphogenetic principles?

Across the digital domain, GPU-based liveness and parameter exploration in *Edge of Chaos* enabled sustained exploration of artificial life systems that model morphogenetic biological systems (Chapter 3). In the material domain, systems and manual-computational workflows exposed clay's 'active' behaviors by aligning computational steps with material practice, calibration, and tacit knowledge (Chapter 4). In hybrid practice, differential growth and agent-based toolpaths bridged these domains, with real-time digital discovery informing materially viable fabrication, and material constraints revealing clay's behavior in addition to its digital construction (Chapter 5).

RQ2. How do interactions between autonomous computational processes and material fabrication practices shape the distribution of creative agency?

Creative agency is situated in design decisions that allocated control across actors. Guiding system evolution using parameter instrumentalization preserved human authorship while amplifying computational agency (Chapter 3). Clay's non-uniform shrinkage, extrusion dynamics, and production risks repositioned agency in material processes, requiring orchestration strategies such as when to let matter speak, when to constrain the physical properties with digital interventions (Chapters 4–5). This redistribution showed itself in design choices such as variable extrusion in *Distributed Agency Trails*, extrusion calibration in *ClayToolkit*, and accepting material 'glitches' as collaborative effects. The resulting artifacts emerged from negotiated relations among human, computational, and material actors rather than any single author.

RQ3. How can new computational systems, techniques, or workflows support emergent expressive practices?

I contributed a set of systems and workflows: a GPU compute framework for real-time artificial life exploration (*Edge of Chaos*) and used this within a large-scale media installation (*Simulacra Naturae*), manual–computational adaptable workflow (*CeramWrap*), toolkit containing various clay 3D printing related components (*ClayToolkit*), and hybrid morphogenetic pipelines (*Morphogenic Clay Vessels*, *Distributed Agency Trails*) that translate algorithmic emergence into clay fabrication toolpaths. Together they function as instruments so practitioners can collaborate with distributed agencies.

6.1.4 Limitations

This practice-led account has several limitations. First, expert-ceramicist evaluation of the hybrid systems was limited as resident artists brought established formal languages not always aligned with emergent digital forms (Chapter 5). Second, although most details come from clay's behaviors, fabrication techniques explained through *Clay-*

Toolkit, *Morphogenic Clay Vessels*, *Distributed Agency Trails* are grounded in PotterBot constraints, and generalization to other materials and machine specifications require adaptation. Third, the full liveness comparison between CPU-based implementations in Chapters 4 and 5 with GPU-based system implementations in Chapter 3 is limited. Fourth, even though agency distribution happened bidirectionally between actors, materials to computational agents require further investigation. Finally, aesthetic judgments of interestingness remain subjective despite public exhibitions and expert critique.

6.2 Future Directions

In this section, I present the detailed component interactions in *Simulacra Naturae*, along with its anticipated future iterations. Finally, I close this dissertation by discussing the potential applications of emergent expression to other material domains and generative AI.

6.2.1 Iterating on *Simulacra Naturae*

This prototype is a data-driven media installation exploring distributed creative agencies through digital generative systems, material ecologies (physical and cyber-physical), and biological computation. This version of *Simulacra Naturae* was deployed in the TransLab as part of *Deep Cuts* 2025, and future plans are presented at the end of current section. I discussed its generative visual components in Chapter 3, and reserved from including in Chapter 5 to keep it within the bounds of generative algorithms and clay agency. However, this work contains nonhuman creative agencies in the same environment (i.e. computational, physical, biological), and its future iterations will explore a more integrated clay fabrication workflow responding to its other layers.

The system is composed of four interwoven layers: the physical environment, cyber-



Figure 6.1: The installation of *Simulacra Naturae*. (Left) Close-up images of (top–to-bottom) the LED matrix; hydroponic plant in glassware; two artifacts from *Morphogenic Clay Vessels*. (Right) Detailed photos of (top-to-bottom) a variation of the projected visuals; one of the islands in the forest-like environment; floor projection on the forest-like environment.

physical interfaces, digital behavioral systems, and the audio layer. In this section, I give brief descriptions of these layers that interact and respond to one another and are driven by the brain organoid data. For complete descriptions, readers can refer to [48]. Their mutual responsiveness co-create a continuously shifting experience that forms a coherent ecology.

Data Source and Processing: *Simulacra Naturae* is driven by a dataset of neuronal activity recorded from human brain organoids. These organoids were cultivated from stem cells and interfaced with electrodes to capture activity, and the resulting raw data were sorted to yield 131 active channel neurons. 131-channel active neurons serve as the primary dataset for this system, and a subset of 27 units, referred to as backbone neurons, demonstrate consistent temporal coordination and strong interactivity. 131 neurons informs the behavior of the digital simulations, and only the backbone subset is mapped onto the cyber-physical system. These physical components resonate syn-

chronously with the live neuron firing data and adds an embodied sonic and kinetic layer to the installation.

Physical Components: The spatial organization of the forest-like environment was directly informed by the neural structure of the brain organoid itself. Specifically, we extracted an abstracted geometry from the firing regions of backbone neurons and mapped to a topological map–clusters of coordinated neural activity that formed functional neighborhoods within the dataset. By calculating the average zones where these backbones formed dense interactions, we derived a set of planar paths and region boundaries, which were then used to shape the layout of plant placement and movement corridors within the installation. This forest-like environment invites visitors into an immersive terrain shaped by neural rhythm and organic materiality. The planting includes large-leaf species such as Monstera deliciosa, Monstrea adansonii, Alocasia, and Strelitzia alba, alongside various hydroponically cultivated plants such as Dracaena sanderiana (bamboos) and Monstrea adansonii (monkey mask). We procured some of these plants before the exhibition, but I have been growing the monkey masks and bamboos over four years. These organisms inhabit a soft substrate composed of mulch and artificial moss, visually resembling the undergrowth and mimics its olfactory and tactile presence. Within this landscape



Figure 6.2: Outcomes of *Morphogenic Clay Vessels* and one example from *Distributed Agency Trails* (lower-left) were used in *Simulacra Naturae* in 2025 as part of its generative ecosystem to highlight both projects' connection to natural formal processes.

are digitally fabricated morphogenic ceramic vessels of varying dimensions, resembling natural forms like coral reefs (Figure 6.2). The system description using a differential growth algorithm and clay fabrication process producing these particular objects have been discussed in detail in Section 5.2.

Cyber-Physical Components: The cyber-physical layer transforms abstract neural rhythms into tactile, acoustic, and luminous phenomena distributed throughout the environment. These components respond, resonate, and participate in the ongoing emergence of the system. Solenoids are embedded in and around the computationally fabricated ceramic vessels and the suspended glassware. These actuators strike the ceramics with precise timing of the organoid firing intensity. The result is a real-time acoustic landscape shaped by collective behavior of micro-strikes, tremors, and pulses that make invisible data physically audible. This makes the vessels 'active', resonating differently depending on the geometry and material thickness. This positions the tactile sound component as co-expression of organic and artificial systems. Two 16x16 RGB LED matrices embedded in custom designed frames drive fiber optics blend with the rest of the environment and introduce a stark contrast between the physical and the cyber entities. The LED matrices mirror the timings of brain organoid firings. Fiber optic strands, embedded in the glassware that carry the hydroplants and around the physical components, carry light into the edges of the installation and create micro-landscapes driven by the neural activations. Together, these components form what Beesley call a 'liminal responsive architecture', a membrane between sensing and sensing back [227]. They allow for material intelligence to emerge through real-time physical response. The space becomes a soft machine in which material and digital become entangled in a continuous negotiation.

Visual Components: The digital visual layer forms a dialogue between the physical and cyber-physical components by constructing a virtual ecosystem of various simulated life-like agents. I discuss these components in Chapter 3 with respect to collaborating with computational agencies and parameter control.

Audio Component: The sound environment is a 16.2-channel generative composition implemented by Diarmid in Max/MSP, driven by neural activity data obtained from brain organoids. Two primary datasets structure the sonic behavior of the installation: population firing rate, governing parameter synthesis and determining soundscape dynamics, and temporal bursting event boundaries, cues harmonic transitions. The primary sampled sound material consists of distorted electric guitar recordings with various treatments, including granulation, pitch shifting, amplitude envelope shaping, feedback delay, spatial distribution, and reverb. These treatments produce two principal sonic textures: sustained guitar tones and granulated fragments. Sustained guitar tone density is determined by population firing rate, resulting in sparse activity at low rates and dense textures at low rates that thin out as activity intensifies. Grain duration is linearly correlated with firing rate, with shorter grains reading samples more quickly and generating upward pitch shifts. Spatialization is randomized per grain. Additional sound sources include sine tone drones and a synthesized low-frequency kick. Sine tones are randomly cued at long attack, sustain, and decay phases, and spatially localized independently. The kick is routed exclusively through subwoofers and functions as a low-end anchor. Each of the 16 output channels features its own dedicated reverb processor, contributing to the immersive spatial character of the installation and modulating all audio routed through it.

System Synchronization & Deployment: We implemented an application in TouchDesigner to parse and synchronize the neural recording data. TouchDesigner functions as the master clock, sequentially reading rows and broadcasting a row index via OSC to all connected subsystems. This index drives the generative behavior of visual, sonic, and physical elements (component diagram is shown in Figure 3.7). Unity and Processing handle real-time visualizations on the surrounding walls, while a second TouchDesigner process runs a real-time AI diffusion pipeline based on Stable Diffusion 2.1, implemented by Iason. This projection is mapped onto the central forest structure. The diffusion model operates using vector images that define the spatial boundaries of the three plant 'islands' on the ground, alongside text prompts that dynamically control visual output. Max/MSP handles spatialized audio and the solenoid-actuated glassware, both of which respond dynamically to the incoming index value. The dataset indices were transmitted over a local area network using OSC from the TouchDesigner patch that concurrently drives the visual component of the installation. Data-sharing architecture ensures real-time synchronization between audio and visual modalities. This architecture enables precise temporal coordination across heterogeneous media outputs, ensuring that all elements respond in synchrony to the unfolding neural dynamics.

The system consisted of two high-performance desktop workstations. One machine was responsible for real-time inference of the AI diffusion model, TouchDesigner-based control logic, sound through MAX/MSP, and the cyber-physical system with the solenoids and the fiber optics. The second managed real-time visual rendering using Unity and Processing. The AI-generated visuals were projected onto the forest structure using a 4K laser floor projector connected to the first machine. The second machine drove a three-projector 4K laser setup arranged in a U-shape configuration, covering two walls. Projection alignment and warping were performed using TouchDesigner's native tools. Frame-accurate synchronization across subsystems was achieved using OSC and for tex-

ture streaming between machines, NDI.

Future Development & Planned Deployment

In the next phase of this project, we intend to integrate neuronal activity (Section 6.2.1) into the generation and fabrication of novel ceramic objects. This approach will enhance the system's demonstration of distributed creative agencies, particularly its manifestation of material agency.

Integrating neuronal rhythms into the generative toolpath used for clay fabrication would make global or temporal parameters of *Morphogenic Clay Vessels* and *Distributed Agency Trails* locally structured and globally varied. Specifically, mapping differential growth's separation-cohesion ratio to the temporal unfolding of neuron bursts would modulate the entire vessel continuously. Similarly, assigning one neuron's activity to one trail agent and mapping the burst activities to behavioral patterns would create unexpected formal outcomes in the toolpath. These 131 toolpath can later be combined in various configurations, or be used as separate artifacts on their own.

These clay forms can be further modified manually by creating holes and spaces for other structures to be incorporated. Composite cyber-artifacts can be created by adding other materials and equipment, such as speakers and actuation mechanisms, to one cyber-artifact. These cyber-artifacts can also be combined with other materials like textile, PLA, acrylic, and metal. For instance, a composite artifact could include a PLA 3D print of termites from the visual simulation, with the internal intricate details displayed through lighting.

This installation is designed to be highly adaptable. The future deployment will consolidate the various component layers described above onto a smaller footprint of (2 m x 2 m) by positioning the new hybrid cyber-artifact in a central location as the driver of the components in *Simulacra Naturae*. Positioned in and around this central

artifact, plants and other physical components will further reinforce the exhibition's ties to simulated natures.

6.2.2 Exploring Other Material Domains

This dissertation primarily explored material agency through clay. In future work, I am interested in investigating other material domains to understand this agency more broadly and whether the practices developed from this research can be applied in different contexts.

Glassworking is a promising site for inquiry because it shares similarities with clay workflows and involves working in high-temperature environments. These settings are inherently unpredictable and, as discussed through insights in *Morphogenic Clay Vessels* and *CeramWrap*, allows for emergence to show itself in material properties and breakages. I am curious how these behaviors could translate to new expressive opportunities when its integrated with computational technologies, and what they might suggest about control and authorship in this domain.

I am also interested in exploring the aesthetics of materials that grow or self-organize, such as bacterial cellulose, mycelium, or plant-guided processes. I have combined my long-standing interest in plant care with simulated life-like ecologies in *Simulacra Naturae*, and I would like to delve deeper into the expressive possibilities of self-organizing material fabrication and computational generative processes. The central question would be how computational constraints and patterns interact with living dynamics, and what forms of documentation can make that interaction understandable.

6.2.3 Extending Emergent Expression to Artificial Intelligence

Artificial intelligence technologies are becoming part of the socio-technological landscape and offer a new form of computational agency. Although its generative power for various media formats is interesting, in the context of emergent expression, I view its connections to software development practices as more important. In my experience, 'traditional' or coding-based approaches to generative art is more akin to craft practices, where the artist is attuned to the medium and engages with its tendencies in a sustained manner. In this space, AI-powered coding approaches empowers human artist and should not be overlooked. For programming-based generative art central to emergent expression, AI-assisted IDEs reduce barriers to implementation and debugging. In the 2025 iteration of *Edge of Chaos* and in second phase of *Distributed Agency Trails*'s implementation, they helped shift my efforts from low-level fixes to higher-level aesthetic steering. As programming approaches natural language, the major barrier of creating hidden structures for emergent behaviors becomes less prominent, enabling the human creative agency to shine not in the low-level system descriptions but in steering aesthetic structures.

AI also accelerates parameter exploration. Beyond faster trial-and-error, new methods for guided search such as learned surrogates and controller tuning, show promise for exploring large spaces and stabilizing complex behaviors [129]. This can support the iterative, open-ended workflows emphasized in this dissertation. In the future, I aim to extend these technologies to improve the perceptual and cognitive aspects of artificial life entities: enabling agents to sense, remember, and predict within their Umwelt so that rules adapt over time. Approaches like learned controllers, world models, or reinforcement learning may allow richer feedback loops that remain steerable at the phase transitions seen in *Edge of Chaos*.

Finally, I plan to examine agentic notions in AI and how they connect to distributed creative agency. I am interested in how to frame AI systems as collaborators within emergent expression so that creative authorship and aesthetic responsibility remain clear while preserving the generativity of multi-actor making.

Bibliography

[1] C. Paul, *Digital Art*. World of Art. Thames & Hudson, London, third edition ed., 2015.

[2] L. Jones, B. Ferran, P. Frank, and T. Funk, *Coded: Art Enters the Computer Age, 1952-1982*. Los Angeles County Museum of Art, Los Angeles, CA, 2023.

[3] M. McCullough, *Abstracting Craft: The Practiced Digital Hand*. 1996.

[4] S. Wolfram, *A New Kind of Science*. 1990.

[5] M. A. Boden, *The Creative Mind: Myths and Mechanisms: Second Edition*. 2003.

[6] M. Mitchell, *Complexity: A Guided Tour*. Oxford University Press, New York, NY, 1. iss. as an paperback ed., 2011.

[7] J. A. Sabloff and P. L. W. Sabloff, *The Emergence of Premodern States: New Perspectives on the Development of Complex Societies*. Seminar Series. SFI Press, Santa Fe (N.M.), 2018.

[8] S. Artut, *Geometric Patterns with Creative Coding*. Apress Berkeley, CA, 2023.

[9] Z.-W. Wu, H. Qu, and K. Zhang, *A Survey of Recent Practice of Artificial Life in Visual Art*, *Artificial Life* **30** (Feb., 2024) 106–135.

[10] O. Bown and J. McCormack, *Creative Agency: A Clearer Goal for Artificial Life in the Arts*, in *European Conference on Artificial Life*, vol. 5778, Jan., 2011.

[11] J. Bennett, *Vibrant Matter*. Duke University Press, 2010.

[12] C. Knappett and L. Malafouris, eds., *Material Agency*. Springer US, Boston, MA, 2008.

[13] M. Toka and S. Balcisoy, *Konumsal Verilerin Fiziksel Gorsellestirmeler ile Incelenmesi*, in *USMOS*, (Ankara, Turkey), 2015.

[14] E. Kaya, M. Toka, A. Bayrak, B. Bozkaya, and S. Balcisoy, *Incorporating Tabletop Visual Analytics into the Decision-making Process: A Case Study of Retail Banking*, in *IEEEVis*, p. 5, 2015.

[15] M. Toka, *Siren: General-Purpose Algorithmic Composition and Live-coding Environment*. PhD thesis, UC Santa Barbara, Santa Barbara, CA, 2018.

[16] C. Ince and M. Toka, *Siren: Hierarchical Composition Interface*, in *Proceedings of the 2017 International Computer Music Conference*, (Shanghai, China), pp. 36–39, Michigan Publishing, 2017. 2027/spo.bbp2372.2017.003.

[17] M. Toka, C. Ince, and M. A. Baytas, *Siren: Interface for Pattern Languages*, in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Virginia Tech), pp. 53–58, 2018.

[18] A. Goodchild, *Emergence and Generative Art*, July, 2022.

[19] F. E. Rosas, B. C. Geiger, A. I. Luppi, A. K. Seth, D. Polani, M. Gastpar, and P. A. M. Mediano, *Software in the natural world: A computational approach to hierarchical emergence*, June, 2024.

[20] T. Tihanyi, *Making and Breaking Rules: Algorithmic Forms and Tactile Processes - A Technoceramist's Adventures with Mathematical Thinking*. 2020.

[21] A. Lomas, *Cellular forms: An artistic exploration of morphogenesis*, in *ACM SIGGRAPH 2014 Studio on - SIGGRAPH '14*, (Vancouver, Canada), pp. 1–1, ACM Press, 2014.

[22] J. McCormack and A. Dorin, *Art, Emergence, and the Computational Sublime*, in *Proceedings of The Second International Conference on Generative Systems in the Electronic Arts*, (Australia), pp. 67–81, Monash University, 2001.

[23] P. Galanter, *What is generative art? Complexity theory as a context for art theory*, *6th Generative Art Conference* (2003) 21–21.

[24] T. Ingold, *The Materials of Life*, in *Making: Anthropology, Archaeology, Art and Architecture*, pp. 17–31. Routledge, London, 2013.

[25] P. Galanter, *An introduction to complexism*, *Technoetic Arts* **14** (June, 2016) 9–31.

[26] B. Latour, *On actor-network theory: A few clarifications*, *Soziale Welt* **47** (1996), no. 4 369–381, [40878163].

[27] G. Simondon, *Individuation in Light of Notions of Form and Information. Volume 1*. No. 57 in Posthumanities. Minnesota Press, Minneapolis London, 2020.

[28] M. McCullough, *Medium*, in *Abstracting Craft: The Practiced Digital Hand*, vol. 11, pp. 193–219. 1998.

[29] A. Nordin, A. Hopf, and D. Motte, *Generative Design Systems for the Industrial Design of Functional Mass Producible Natural-Mathematical Forms*, .

[30] L. Barbieri and M. Muzzupappa, *Form innovation: Investigating the use of generative design tools to encourage creativity in product design*, *International Journal of Design Creativity and Innovation* **0** (2024), no. 0 1–20.

[31] LIA, "Filament sculptures." `URL`, 2013.

[32] M. MacDowell and J. Avery, *Craft Works! Michigan: A Report on Traditional Crafts and Economic Development in Michigan*, tech. rep., Michigan State University Museum, Dec., 2006.

[33] P. Adler and T. Osman, *California's creative economy: Key highlights from 2023*, tech. rep., Otis College of Art and Design, Los Angeles, 2024.

[34] J. Mitchell and G. Joyce, *Building on the Past, Facing the Future : Renewing the Creative Economy of New Mexico*. University of New Mexico, Bureau of Business & Economic Research ; New Mexico Department of Cultural Affairs, Albuquerque, N.M., 2014.

[35] U. N. E. Network, "New Economics for Sustainable Development: Creative Economies." `URL`, 2021.

[36] J. McCormack, *Working with Generative Systems: An Artistic Perspective*, in *Electronic Visualisation and the Arts (EVA 2017)*, BCS Learning & Development, July, 2017.

[37] M. A. Boden and E. A. Edmonds, *What is generative art?*, *Digital Creativity* **20** (2009), no. 1-2 21–46.

[38] P. Galanter, *Generative Art Theory*, in *A Companion to Digital Art*, ch. 5, pp. 146–180. John Wiley & Sons, Ltd, 2016.

[39] J. McCormack, O. Bown, A. Dorin, J. Mccabe, G. Monro, and M. Whitelaw, *Ten Questions Concerning Generative Computer Art*, tech. rep., 2012.

[40] P. Galanter, *Computational Aesthetic Evaluation: Past and Future*, in *Computers and Creativity* (J. McCormack and M. d'Inverno, eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[41] C. G. Langton, *Computation at the edge of chaos: Phase transitions and emergent computation*, *Physica D: Nonlinear Phenomena* **42** (June, 1990) 12–37.

[42] M. Toka, *Creative Computational Systems at the Edge of Chaos*, tech. rep., Sept., 2021.

[43] Arsiliath, "Arsiliath on Instagram." URL, 2025. Accessed: 2025-06-04.

[44] M. Toka, *The edge of chaos*, in *Hybrid Science Experimentation*, (Barcelona, Spain (online)), EXP. Experimental Photo Festival, 2021. Group exhibition curated by Felicita Russo and Maciej Zapiór.

[45] M. Toka, *The edge of chaos*, in *SyMADES '22*, (Santa Barbara, CA, USA), California NanoSystems Institute, 6, 2022. Group exhibition curated by Marko Peljhan.

[46] M. Toka, *The edge of chaos*, in *Deep Cuts '25*, (Santa Barbara, CA, USA), California NanoSystems Institute, 6, 2025. Group exhibition curated by researchers in Media Arts & Technology.

[47] M. Toka, N. Manoudaki, I. Paterakis, and D. Flatley, *Simulacra natura*, in *Deep Cuts '25*, (Santa Barbara, CA, USA), California NanoSystems Institute, 6, 2025. Group exhibition curated by researchers in Media Arts & Technology.

[48] N. Manoudaki, M. Toka, I. Paterakis, and D. Flatley, *Simulacra naturae: Generative ecosystem driven by agent-based simulations and brain organoid collective intelligence*, in *2025 IEEE VIS Arts Program (VISAP)*, 2025-10.

[49] M. Toka, D. Frost, S. Bourgault, A. Farber, C. Friedman-Gerlicz, R. Lee, E. Paek, P. Wiley, and J. Jacobs, *Practice-driven Software Development: A Collaborative Method for Digital Fabrication Systems Research in a Residency Program*, in *Designing Interactive Systems Conference*, (IT University of Copenhagen Denmark), pp. 1192–1217, ACM, July, 2024.

[50] M. Toka, S. Bourgault, C. Friedman-Gerlicz, and J. Jacobs, *An Adaptable Workflow for Manual-Computational Ceramic Surface Ornamentation*, in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, (New York, NY, USA), pp. 1–15, Association for Computing Machinery, Oct., 2023.

[51] T. De Wolf and T. Holvoet, *Emergence Versus Self-Organisation: Different Concepts but Promising When Combined*, in *Engineering Self-Organising Systems* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. A. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, and R. Nagpal, eds.), vol. 3464, pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[52] J. Wagemans, J. Feldman, S. Gepshtein, R. Kimchi, J. R. Pomerantz, P. A. Van Der Helm, and C. Van Leeuwen, *A century of Gestalt psychology in visual perception: II. Conceptual and theoretical foundations.*, *Psychological Bulletin* **138** (2012), no. 6 1218–1252.

[53] L. Von Bertalanffy, *General System Theory: Foundations, Development, Applications*. 1968.

[54] D. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*. 1999.

[55] K. O. Bayraktar, *Sistem Teorisi Baglaminda Sanat Nesnesi ve Esleme*. PhD thesis, 2017.

[56] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine*. Oct., 2019.

[57] D. Krakauer, ed., *Foundational Papers in Complexity Science*. SFI Press, Santa Fe, 2024.

[58] D. Krakauer, *The Complex World: An Introduction to the Foundations of Complexity Science*. The SFI Press Scholars Series, Santa Fe, NM, Sept., 2024.

[59] A. Turing, *The Chemical Basis of Morphogenesis*, .

[60] A. T. Winfree, *The prehistory of the Belousov-Zhabotinsky oscillator*, *Journal of Chemical Education* **61** (Aug., 1984) 661.

[61] J. Gleick, *Chaos: Making a New Science*. 2011.

[62] M. Cook, *Universality in Elementary Cellular Automata*, *Complex Systems* (2004) 40.

[63] M. Gardner, *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"*, *Scientific American* (Oct., 1970) 120–123.

[64] M. Heidegger, J. Macquarrie, and E. Robinson, *Being and Time*. Blackwell publ, Malden (Mass.), 2007.

[65] R. Morris, *Anti Form*, in *Continuous Project Altered Daily: The Writings of Robert Morris*, pp. 41–47. MIT Press ; Solomon R. Guggenheim Museum, Cambridge, Mass. : New York, N.Y, 1993.

[66] R. Morris, "Robert Morris - Untitled (Pink Felt) - The Guggenheim Museum." `URL`, 2025. Accessed: 2025-08-25.

[67] P. Galanter, *What is Complexism? Generative Art and the Cultures of Science and the Humanities*, in *Proceedings of International Conference on Generative Art*, 2008.

[68] J. A. Hoelscher, *Art as Information Ecology: Artworks, Artworlds, and Complex Systems Aesthetics (Thought in the Act)*. Duke University Press, Oct., 2021.

[69] U.-M. O'Reilly, I. Ross, and P. Testa, *Emergent Design: Artificial Life for Architecture Design*, in *Artificial Life VII* (M. A. Bedau, J. S. McCaskill, N. H. Packard, and S. Rasmussen, eds.), pp. 454–464. The MIT Press, Aug., 2000.

[70] J. B. Longo, *Emergent Making: Material Expression, Character, Performance.* Master of Science, Drexel University, Philadelphia, Pennsylvania, United States, 2019.

[71] D. Bennett, A. Dix, P. Eslambolchilar, F. Feng, T. Froese, V. Kostakos, S. Lerique, and N. Van Berkel, *Emergent Interaction: Complexity, Dynamics, and Enaction in HCI*, in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, (New York, NY, USA), pp. 1–7, Association for Computing Machinery, May, 2021.

[72] W. Gaver, P. G. Krogh, A. Boucher, and D. Chatting, *Emergence as a Feature of Practice-based Design Research*, in *Designing Interactive Systems Conference*, (Virtual Event Australia), pp. 517–526, ACM, June, 2022.

[73] V. Douchová, *Birkhoff's aesthetic measure*, *Acta Universitatis Carolinae Philosophica et Historica* **XXI** (2015), no. 1 39–53.

[74] G. W. Smith, *An Interview with Frieder Nake*, *Arts* **8** (May, 2019) 69–69.

[75] A. Moles, *Information Theory and Esthetic Perception.* University Illinois Press, 1969.

[76] F. Nake, *Information Aesthetics: An heroic experiment*, *Journal of Mathematics and the Arts* **6** (June, 2012) 65–75.

[77] S. Audry, *Art in the Age of Machine Learning.* The MIT Press, Cambridge, Massachusetts, 2021.

[78] M. Amerika, "An Interview with Frieder Nake." `URL`. Accessed: 2025-06-24.

[79] J. Reichardt, ed., *Cybernetic Serendipity: The Computer and the Arts.* Frederik A. Praeger, New York; Washington, 2018.

[80] T. Ropac, "Vera molnár molndrian, 74,066/13.36.22, 1974." `URL`. Accessed: 2025-06-24.

[81] F. Nake, *Georg Nees & Harold Cohen: Re:tracing the origins of digital media*, in *Digital Art through the Looking Glass. New Strategies for Archiving, Collecting and Preserving in Digital Humanities.* 2019.

[82] F. Nake, *Someone is always already there in front of you even though you may not like it*, *Journal of Science and Technology of the Arts* **10** (Sept., 2018) 2–15–2–25.

[83] P. Weibel, S. Donderer, and ZKM Zentrum für Kunst und Medien, eds., *BioMedia: The Age of Media with Life-like Behavior*. A ZKM Book. Spector Books, Leipzig, 2023.

[84] G. Pask, *A comment, a case history and a plan*, .

[85] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press, 1992.

[86] N. Collins, *The Analysis of Generative Music Programs*, *Org. Sound* **13** (Dec., 2008) 237–248.

[87] R. Pieters and S. Winiger, "Creative AI: On the Democratisation & Escalation of Creativity — by creative.ai — Medium." URL, Mar., 2016.

[88] M. Ide, "Generative UI." URL, 2015.

[89] S. Chung, "Projects." URL, 2024.

[90] M. A. Boden, *Computer models of creativity*, *AI Magazine* **30** (2009), no. 3 23–34.

[91] R. Anadol, "Projects." URL, 2024.

[92] J. Tarbell, "Complexification." URL, 2012.

[93] A. Hoff, "inconvergent." URL, 2024.

[94] M. Cooper, "Works." URL, 2024.

[95] C. Raes, "Works." URL, 2024.

[96] A. Noë, *Strange Tools*. 2015.

[97] V. S. Ramachandran, *A Brief Tour of Human Consciousness: From Impostor Poodles to Purple Numbers*. Pi Press, New York, 2005.

[98] Y. Bo, J. Yu, and K. Zhang, *Computational aesthetics and applications*, *Visual Computing for Industry, Biomedicine, and Art* **1** (Dec., 2018) 6.

[99] K. Sims, *Artifical Evolution for Computer Graphics*, in *SIGGRAPH*, (Las Vegas, NV), 1991.

[100] R. Falconer, "Alternative Evolution — William Latham." URL. Accessed: 2025-06-24.

[101] C. G. Langton, *Artificial Life*, in *ARS Electronica Catalog*, 1993.

[102] Aristotle, "Aristotle on technology and nature." URL, 1999. Accessed: 2024-06-07; Translated excerpts and commentary compiled by Joachim Schummer.

[103] C. W. Reynolds, *Flocks, Herds, and Schools: A Distributed Behavioral Model*, *Computer Graphics (ACM)* **21** (Aug., 1987) 25–34.

[104] J. Jones, *Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks*, *Artificial Life* **16** (Apr., 2010) 127–153.

[105] B. W.-C. Chan, *Lenia and Expanded Universe*, *The 2020 Conference on Artificial Life* (2020) 221–229, [arXiv:2005.0374].

[106] B. W.-C. Chan, *Lenia - Biology of Artificial Life*, *Complex Systems* **28** (Oct., 2019) 251–286, [arXiv:1812.0543].

[107] L. Caussan and A. Sawyer, "The bibites." URL, 2025. Accessed: 2025-08-09.

[108] C. Heinemann, "Alien - explore worlds of artificial life." URL, 2025. Accessed: 2025-08-09.

[109] Emergent Garden, "Life engine." URL, 2025. Accessed: 2025-08-09.

[110] V. Todorovic, *Reimagining life (forms) with generative and bio art*, .

[111] MoMA, "Ian cheng: Emissaries." URL, 2017. Accessed: 2025-08-09.

[112] Spacefiller, "Living walls." URL, 2025. Accessed: 2025-08-09.

[113] P. Whidden, "Mote: An interactive ecosystem simulation." URL, 2025.

[114] C. Sommerer and L. Mignonneau, *Living Poetry*, *Artificial Life* **21** (Aug., 2015) 313–319.

[115] S. Crespo, "neural zoo." URL, 2022. Accessed: 2025-08-09.

[116] H. H. Ji and G. Wakefield, *Entanglement: an immersive art of an engagement with non-conscious intelligence*, in *ISEA2025: 30th International Symposium on Electronic Art – Exhibition Catalogue* (S. W. Roh and Y. H. Roh, eds.). Art Center Nabi, 2025.

[117] E. Kac, *Bio art*, .

[118] S. Anker, *Epistemic practices in bio art*, .

[119] T. Hobbs, "The Rise of Long-Form Generative Art." URL, Aug., 2021.

[120] S. Crispin, "Mat seminar: Onchain generative art: The world computer vs the casino." URL, 2025. Accessed: 2025-08-09.

[121] T. Hobbs, "Reflections on Five Years of Making Art Through Programming." URL, May, 2019.

[122] T. Hobbs, "On Sketching (and the Life-Cycle of Artistic Ideas)." URL, Aug., 2023.

[123] T. Hobbs, "My Artistic Process: Haecceity Series." URL, Dec., 2014.

[124] A. Lomas, *Species Explorer: An interface for artistic exploration of multi-dimensional parameter spaces*, in *Electronic Visualisation and the Arts*, 2016.

[125] A. Saunier, Salter, Chris, Vermette, Julien, Quessy, Alexandre, Demeule, Simon, d'Ark, Ursula J'vlyn, Jain, Puneet, and Audry, Sofian, *autonomX—Real Time Creation/Composition with Complex Systems*, in *Proceedings of International Symposium of Electronic Arts*, (Barcelona, Spain), pp. 540–547, 2022.

[126] K. Youn, *A Compositional Interface for Generative Audiovisual Systems*. PhD thesis, UCSB, 2019.

[127] C. R. Shalizi and K. L. Shalizi, *Quantifying self-organization in cyclic cellular automata*, in *Noise in Complex Systems and Stochastic Dynamics* (L. Schimansky-Geier, D. Abbott, A. Neiman, and C. Van den Broeck, eds.), vol. 5114, pp. 108–108, SPIE, May, 2003.

[128] H. V. D. Parunak, P. O. Box, A. Arbor, S. Brueckner, P. O. Box, and A. Arbor, *Entropy and Self-Organization in Multi-Agent Systems*, in *Proceedings of the International Conference on Autonomous Agents*, no. Agents, pp. 124–130, 2001.

[129] A. Kumar, C. Lu, L. Kirsch, Y. Tang, K. O. Stanley, P. Isola, and D. Ha, *Automating the Search for Artificial Life with Foundation Models*, May, 2025.

[130] H. H. Glassie, *Turkish Traditional Art Today*. Indiana University Press, Bloomington, 1st ed. edition ed., Dec., 1993.

[131] F. H. Harlow, *Modern Pueblo Pottery: 1880-1960*. Northland Press / Flagstaff, 1977.

[132] L. Buechley and R. Ta, *3D Printable Play-Dough: New Biodegradable Materials and Creative Possibilities for Digital Fabrication*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), pp. 1–15, Association for Computing Machinery, Apr., 2023.

[133] L. Buechley, J. Gould, and F. Bell, *CeraMetal: A New Approach to Low-Cost Metal 3D Printing with Bronze Clay*, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, (New York, NY, USA), pp. 1–16, Association for Computing Machinery, May, 2024.

[134] C. Friedman-Gerlicz, J. Gould, F. Bell, and L. Buechley, *ColdGlass: Full-Color Desktop 3D Printing in Glass*, in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), pp. 1–20, Association for Computing Machinery, Apr., 2025.

[135] F. Bell, C. Friedman-Gerlicz, L. Urenda, and L. Buechley, *3D Printing Eggshells: Exploring Eco-Socio-Technical Relations through Biomaterial Design*, in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, (New York, NY, USA), pp. 1–19, Association for Computing Machinery, Apr., 2025.

[136] F. Bell, E. Mcclure, C. Friedman-Gerlicz, R. Ta, and L. Buechley, *Shape-Changing Clay-Dough: Taking a Material-Oriented Approach to 3D Printing Ceramic Forms*, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, (New York, NY, USA), pp. 1–19, Association for Computing Machinery, May, 2024.

[137] S. Bourgault, *Developing Action-Oriented Systems for Manual-Computational Craft Workflows*, in *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, (San Francisco CA USA), pp. 1–5, ACM, Oct., 2023.

[138] L. Devendorf and K. Ryokai, *Being the machine: Reconfiguring agency and control in hybrid fabrication*, *Conference on Human Factors in Computing Systems - Proceedings* **2015-April** (2015) 2477–2486.

[139] K. Andersen, R. Wakkary, L. Devendorf, and A. McLean, *Digital crafts-machine-ship: Creative collaborations with machines*, *Interactions* **27** (Dec., 2019) 30–35.

[140] A. Cheatle and S. J. Jackson, *Digital entanglements: Craft, computation and collaboration in fine art furniture production*, *CSCW 2015 - Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing* (2015) 958–968.

[141] J. Jacobs and A. Zoran, *Hybrid practice in the kalahari: Design collaboration through digital tools and hunter-gatherer craft*, *Conference on Human Factors in Computing Systems - Proceedings* **2015-April** (2015) 619–628.

[142] Potterbot, "Ceramic 3D Printers." `URL`, 2024.

[143] Vormvrij, "Lutum 3D Ceramics." `URL`, 2024.

[144] J. Keep, *A Guide to Clay 3D Printing*, tech. rep., 2020.

[145] A. Desjardins and T. Tihanyi, *ListeningCups: A Case of Data Tactility and Data Stories*, in *Proceedings of the 2019 on Designing Interactive Systems Conference*, (San Diego CA USA), pp. 147–160, ACM, June, 2019.

[146] A. Desjardins, T. Tihanyi, F. El Shabazz-Thompson, B. Craft, and J. Saimo, *The Inner Ear: Capturing And Physicalizing Home Vibrations*, in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, (New York, NY, USA), pp. 594–607, Association for Computing Machinery, July, 2023.

[147] N. Dick, N. Glauber, A. Yehezkeli, M. Mizrahi, S. Reches, M. Ben-Yona, A. Carmi, and A. Zoran, *Design with Minimal Intervention: Drawing with Light and Cracks*, in *Proceedings of the 2018 Designing Interactive Systems Conference*, (Hong Kong China), pp. 1107–1120, ACM, June, 2018.

[148] C. Zheng, B. Han, X. Liu, L. Devendorf, H. Tan, and C. C. Yen, *Crafting Interactive Circuits on Glazed Ceramic Ware*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), pp. 1–18, Association for Computing Machinery, Apr., 2023.

[149] Lesley, *The 8 Stages of Clay - An Amazing Journey from Dirt to Art*, Nov., 2022.

[150] A. Reijnders, *The Ceramic Process: A Manual and Source of Inspiration for Ceramic Art and Design*. University of Pennsylvania Press, Incorporated, Philadelphia, PA, 2005.

[151] Lesley, *What Happens to Clay When it is Fired in a Kiln?*, Feb., 2021.

[152] L. Bloomfield, *Firing Clay: The Lowdown on the Ceramic Firing Process*, 2017.

[153] L. Buechley, A. Bustos, E. Edreva, T. Fenske, R. Fresquez, S. Hafer, A. Handey, M. Louie, J. Ng, A. Shen, R. Smith, A. Sustaita, M. Truong, K. Vallon, and K. M. Weiss, *A Computational Fabrication Course: Exploring Philosophical Reflection, Real-World Use, Personal Expression, and Social Connection*, in *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, SCF '22, (New York, NY, USA), pp. 1–13, Association for Computing Machinery, Oct., 2022.

[154] M. Hirsch, G. Benabdallah, J. Jacobs, and N. Peek, *Nothing Like Compilation: How Professional Digital Fabrication Workflows Go Beyond Extruding, Milling, and Machines*, *ACM Transactions on Computer-Human Interaction* (July, 2023).

[155] B. Subbaraman and N. Peek, *P5.fab: Direct Control of Digital Fabrication Machines from a Creative Coding Environment*, in *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, DIS '22, (New York, NY, USA), pp. 1148–1161, Association for Computing Machinery, June, 2022.

[156] J. Tran O'Leary, E. Jun, and N. Peek, *Improving Programming for Exploratory Digital Fabrication with Inline Machine Control and Styled Toolpath Visualizations*, in *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, SCF '22, (New York, NY, USA), pp. 1–12, Association for Computing Machinery, Oct., 2022.

[157] H. Twigg-Smith and N. Peek, *Dynamic Toolchains: Software Infrastructure for Digital Fabrication Workflows*, in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, (New York, NY, USA), pp. 1–20, Association for Computing Machinery, Oct., 2023.

[158] F. H. Fossdal, V. Nguyen, R. Heldal, C. L. Cobb, and N. Peek, *Vespidae: A Programming Framework for Developing Digital Fabrication Workflows*, in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, (New York, NY, USA), pp. 2034–2049, Association for Computing Machinery, July, 2023.

[159] M. Horn, A. Traylor, and L. Buechley, *Slabforge: Design Software for Slab-Based Ceramics*, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), pp. 1–12, Association for Computing Machinery, Apr., 2022.

[160] S. Bourgault, P. Wiley, A. Farber, and J. Jacobs, *CoilCAM: Enabling Parametric Design for Clay 3D Printing Through an Action-Oriented Toolpath Programming System*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), pp. 1–16, Association for Computing Machinery, Apr., 2023.

[161] D. Frost, E.-H. Paek, R. Lee, and J. Jacobs, *SketchPath: Using digital drawing to integrate the gestural qualities of craft in CAM-Based clay 3D printing*, in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, (New York, NY, USA), Association for Computing Machinery, 2024.

[162] C. Friedman-Gerlicz, D. Gelosi, F. Bell, and L. Buechley, *WeaveSlicer: Expanding the Range of Printable Geometries in Clay*, in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, (New York, NY, USA), pp. 1–16, Association for Computing Machinery, May, 2024.

[163] S. Hsueh, M. C. Felice, S. F. Alaoui, and W. E. Mackey, *What Counts as 'Creative' Work? Articulating Four Epistemic Positions in Creativity-Oriented HCI Research*, in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, (New York, NY, USA), Association for Computing Machinery, 2024.

[164] D. Bennett, O. Metatla, A. Roudaut, and E. D. Mekler, *How does HCI Understand Human Agency and Autonomy?*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), pp. 1–18, Association for Computing Machinery, Apr., 2023.

[165] M. Dzhimova and F. Tigre Moura, *Calculated Randomness, Control and Creation: Artistic Agency in the Age of Artificial Intelligence*, *Arts* **13** (Oct., 2024) 152.

[166] S. Penny, *Emergence, Agency, and Interaction—Notes from the Field*, *Artificial Life* **21** (Aug., 2015) 271–284.

[167] M. A. Boden, *Agents and Creativity*, *Communications of the ACM* **37** (1994), no. 7 117–121.

[168] M. A. Boden, *Creativity and Art: Three Roads to Surprise*. Oxford University Press, Oxford ; New York, 2010.

[169] J. McCormack and M. d'Inverno, eds., *Computers and Creativity*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[170] J. McCormack and A. Lomas, *Deep Learning of Individual Aesthetics*, *arXiv:2009.12216 [cs]* (Sept., 2020) [arXiv:2009.1221].

[171] J. McCormack and C. Cruz Gambardella, *Complexity and aesthetics in generative and evolutionary art*, *Genetic Programming and Evolvable Machines* **23** (Dec., 2022) 535–556.

[172] C. G. Johnson, J. McCormack, I. Santos, and J. Romero, *Understanding Aesthetics and Fitness Measures in Evolutionary Art Systems*, *Complexity* **2019** (2019), no. 1 3495962.

[173] B. Shneiderman, *Creativity support tools: accelerating discovery and innovation*, *Commun. ACM* **50** (dec, 2007) 20–32.

[174] B. Shneiderman, *Creativity Support Tools*, *Communications of the ACM* **45** (2002), no. 10 116–120.

[175] B. Shneiderman, G. Fischer, M. Czerwinski, M. Resnick, B. Myers, L. Candy, E. Edmonds, M. Eisenberg, E. Giaccardi, T. Hewett, P. Jennings, B. Kules, K. Nakakoji, J. Nunamaker, R. Pausch, T. Selker, E. Sylvan, and M. Terry, *Creativity Support Tools: Report From a U.S. National Science Foundation Sponsored Workshop*, *International Journal of Human–Computer Interaction* **20** (May, 2006) 61–77.

[176] C. Remy, L. MacDonald Vermeulen, J. Frich, M. M. Biskjaer, and P. Dalsgaard, *Evaluating Creativity Support Tools in HCI Research*, in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, (New York, NY, USA), pp. 457–476, Association for Computing Machinery, July, 2020.

[177] J. Kato, *On the Relationship between HCI Researchers and Creators—Or How I Became a Toolsmith*, *XRDS: Crossroads, The ACM Magazine for Students* **29** (June, 2023) 26–31.

[178] J. Kato, J. Frich, Z. Lu, J. Jacobs, K. Nakakoji, and C. Latulipe, *Special Interest Group on Creativity and Cultures in Computing*, in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI EA '23, (New York, NY, USA), pp. 1–4, Association for Computing Machinery, Apr., 2023.

[179] L. Buechley and B. M. Hill, *LilyPad in the wild: How hardware's long tail is supporting new engineering and design communities*, *DIS 2010 - Proceedings of the 8th ACM Conference on Designing Interactive Systems* (2010) 199–207.

[180] G. Fischer, *Social creativity: Turning barriers into opportunities for collaborative design*, *PDC 2004 - Proceedings of the Eight Participatory Design Conference 2004 - Artful Integration: Interweaving Media, Materials and Practices* (2004) 152–161.

[181] L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, N. Shireen, and M. Terry, *GEM-NI: A system for creating and managing alternatives in generative design*, *Conference on Human Factors in Computing Systems - Proceedings* **2015-April** (2015) 1201–1210.

[182] J. Frich, L. MacDonald Vermeulen, C. Remy, M. M. Biskjaer, and P. Dalsgaard, *Mapping the Landscape of Creativity Support Tools in HCI*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), pp. 1–18, Association for Computing Machinery, May, 2019.

[183] J. Li, E. Rawn, J. Ritchie, J. Tran O'Leary, and S. Follmer, *Beyond the Artifact: Power as a Lens for Creativity Support Tools*, in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, (New York, NY, USA), pp. 1–15, Association for Computing Machinery, Oct., 2023.

[184] G. Levin, "For Us, By Us." `URL`, Mar., 2015.

[185] B. Fry and C. Reas, "Processing." `URL`, 2024.

[186] L. Lee McCarthy, "p5.js." `URL`, 2024.

[187] J. Li, S. Hashim, and J. Jacobs, *What We Can Learn From Visual Artists About Software Development*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), pp. 1–14, Association for Computing Machinery, May, 2021.

[188] K. Barad, *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press, Durham London, 2007.

[189] D. Krakauer and C. Kempes, *Is life a complex computational process?*, Sept., 2024.

[190] L. Malafouris, *The Material-Engagement Approach: A Summary of the Argument*, in *How Things Shape the Mind: A Theory of Material Engagement*. The MIT Press, Cambridge, Massachusetts, 2013.

[191] P. L. March, *Playing with clay and the uncertainty of agency. A Material Engagement Theory perspective*, *Phenomenology and the Cognitive Sciences* **18** (Feb., 2019) 133–151.

[192] K. S. Poulsgaard, *Enactive individuation: Technics, temporality and affect in digital design and fabrication*, *Phenomenology and the Cognitive Sciences* **18** (Feb., 2019) 281–298.

[193] T. Ingold, *The textility of making*, *Cambridge Journal of Economics* **34** (2009), no. 1 91–102.

[194] T. Ingold, *Toward an Ecology of Materials*, *Annual Review of Anthropology* **41** (Oct., 2012) 427–442.

[195] P. Baudisch and S. Mueller, *Personal Fabrication*, *Foundations and Trends® in Human–Computer Interaction* **10** (2017), no. 3–4 165–293.

[196] I. Moyer, *Personal fabrication: From automated machines to augmented tools*, *XRDS: Crossroads, The ACM Magazine for Students* **22** (2016), no. 3 28–31.

[197] G. Simondon, *On the Mode of Existence of Technical Objects*, *Deleuze Studies* **5** (Nov., 2011) 407–424.

[198] G. Simondon, *The Genesis of the Individual*, in *Incorporations* (J. Crary and S. Kwinter, eds.). 1992.

[199] S. Mills, *Gilbert Simondon: Information, Technology and Media*. 2016.

[200] T. Roberts, *Thinking technology for the Anthropocene: Encountering 3D printing through the philosophy of Gilbert Simondon*, *cultural geographies* **24** (Oct., 2017) 539–554.

[201] B. Latour, *Reassembling the Social: An Introduction to Actor-Network-Theory*. Clarendon Lectures in Management Studies. Oxford University Press, Oxford ; New York, 2005.

[202] L. Suchman, *Human-Machines Reconfigurations: Plans and Situated Actions*. Cambridge University Press, New York, 2nd ed ed., 2009.

[203] N. Oxman, C. Ortiz, F. Gramazio, and M. Kohler, *Material ecology*, *Computer-Aided Design* **60** (Mar., 2015) 1–2.

[204] K. W. Song and E. Paulos, *Unmaking: Enabling and Celebrating the Creative Material of Failure, Destruction, Decay, and Deformation*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, (Yokohama Japan), pp. 1–12, ACM, May, 2021.

[205] A. Lomas, *Morphogenetic Vase Forms*, in *ALIFE 2019: The 2019 Conference on Artificial Life*, pp. 523–530, July, 2019.

[206] J. Rosenkrantz and J. Louis-Rosenberg, "Floraforms." URL, 2025.

[207] P. Oskam and M. Latour, "Urban reef." URL, 2025. Accessed: 2025-08-12.

[208] L. Candy and E. A. Edmonds, *Practice-Based Research in the Creative Arts*, *Leonardo* **51** (2018), no. 2 111–117.

[209] U. Technologies, "Unity real-time development platform." URL, 2025.

[210] T. Van Der Molen, A. Spaeth, M. Chini, S. Hernandez, G. A. Kaurala, H. E. Schweiger, C. Duncan, S. McKenna, J. Geng, M. Lim, J. Bartram, A. Dendukuri, Z. Zhang, J. Gonzalez-Ferrer, K. Bhaskaran-Nair, L. J. Blauvelt, C. R. Harder, L. R. Petzold, D.-M. Alam El Din, J. Laird, M. Schenke, L. Smirnova, B. M. Colquitt, M. A. Mostajo-Radji, P. K. Hansma, M. Teodorescu, A. Hierlemann, K. B. Hengen, I. L. Hanganu-Opatz, K. S. Kosik, and T. Sharf, *Protosequences in brain organoids model intrinsic brain states*, Dec., 2023.

[211] N. Manoudaki, I. Paterakis, D. Flatley, R. Millett, and M. Novak, *Organoid_protonoesis ii*, in *ISEA2025: 30th International Symposium on Electronic Art – Exhibition Catalogue* (S. W. Roh and Y. H. Roh, eds.), pp. 54–55. Art Center Nabi, 2025.

[212] M. Toka, "Viscousvoid on Instagram." URL, 2025. Accessed: 2025-07-15.

[213] R. Fisch, J. Gravnar, and D. Griffeath, *Threshold-range scaling of cellular automata excitable*, tech. rep., 1991.

[214] T. Lucretius Carus, *The Nature of Things*. Norton, New York, 1st ed ed., 1977.

[215] T. Sharf, T. van der Molen, S. M. K. Glasauer, E. Guzman, A. P. Buccino, G. Luna, Z. Cheng, M. Audouard, K. G. Ranasinghe, K. Kudo, S. S. Nagarajan, K. R. Tovar, L. R. Petzold, A. Hierlemann, P. K. Hansma, and K. S. Kosik, *Functional neuronal circuitry and oscillatory dynamics in human brain organoids*, 2022-07-29.

[216] Derivative, "Touchdesigner." URL, 2025.

[217] S. Davidson, "Grasshopper." URL, 2007.

[218] S. Takada and K. Takada, "Smooth and Exacting Surface Decoration." URL, 2017. Accessed: 2023-04-05.

[219] K. Sing, "Turn studio." URL, 2023. Accessed: 2023-04-04.

[220] E. W. Weisstein, ""developable surface." from mathworld–a wolfram web resource.." URL, 2023. Accessed: 2023-07-20.

[221] N. M. Patrikalakis, T. Maekawa, and W. Cho, "Differential geometry of developable surfaces." URL, 2009. Accessed: 2023-07-20.

[222] alantai, "Food4rhino selectable preview 0.2." URL, 2019. Accessed: 2019-06-08.

[223] E. Objects, "Potterware." URL, 2024.

[224] S. Bakker, S. Gutierrez H, J. Seong, and N. Williams, *Differential Growth: An Exploration*, CSE 6730: Modeling and Simulation 13, Georgia Institute of Technology, Apr., 2022.

[225] J. Webb, *Exploring 2D differential growth with JavaScript*, Oct., 2020.

[226] M. Akten, "Memo akten website." URL, 2025. Accessed: 2025-08-20.

[227] P. Beesley, H. Isaacs, P. Ohrstedt, and R. Gorbet, eds., *Hylozoic Ground: Liminal Responsive Architecture: Philip Beesley*. Riverside Architectural Press, Cambridge, Ont., first edition ed., 2010.